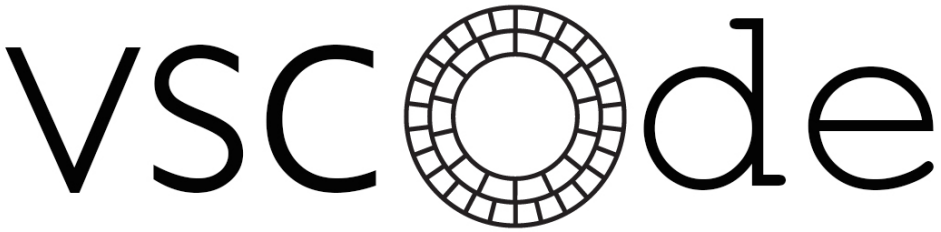


VSCOde Final Report

Jessica Cheng (jc4687)
Anna Lu (ajl2256)
Hana Mizuta (hm2694)
Spencer Yen (ssy2121)
Kenny Yuan (kky2114)

December 19, 2018

Contents

1 Introduction

VSCOde is a language designed to analyze and manipulate images. While our language was initially inspired by the high definition edge-detection technology used by autonomous vehicles, our language can be used to create modern art, perhaps one day rivaling AI art at Christie's that sold for $432,500 in late October.

VSCOde performs various image manipulations, such as color adjustments, basic edge detection, and custom filters. Images are represented as matrices, granting users greater control over the individual pixels in an image and allowing easy application of filters through matrix transformations.

With its matrix-focused nature, VSCOde syntax draws from C, Matlab, and Python.

2 Tutorial

It is recommended to run VSCOde on VirtualBox. Simply follow the VirtualBox Setup Instructions to get started. With VirtualBox set up, install OpenCV and run the program.

2.1 Set Up Homebrew

To ease the installation process, download Homebrew package manager. Run the following command in your command line:

```
/usr/bin/ruby -e "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

Also install Homebrew Cask, a Homebrew extension to speed up OS X installations.

```
brew tap caskroom/cask
```

2.2 Set Up VirtualBox

To quickly get started, install VirtualBox.

```
brew cask install virtualbox
```

Then, download Numel, a Virtual Image with LLVM made by the John Hui.

```
wget http://www.cs.columbia.edu/~sedwards/classes/2018/4115-fall/numel.ova
```

Username/Password: al / plt

2.3 Install OpenCV

OpenCV is a C++ library, used for image manipulation, image loading, and image saving.

```
sudo apt-get install opencv-data
```

Also install the development files for opencv.

```
sudo apt-get install libopencv-dev
```

2.4 Install Clang and LLVM

Clang and LLVM will compile and link in OpenCV.

```
sudo apt install clang-4.0
sudo apt install llvm-5.0
```

2.5 Run Test Cases

Running VSCOde `${basename}.vsc` run the following:

```
make clean
make
./link.sh ${basename}
```

To instead run the pre-written test codes, run the following:

```
./testall.sh
```

3 Language Reference Manual

3.1 Types

3.1.1 Primitives

Type	Description
`int`	32-bit signed integer
`double`	64-bit float point number
`bool`	8-bit boolean variable
`string`	Array of ASCII characters

3.1.2 Structures

Structure	Description	Syntax

`matrix`	Immutable data structure that stores a collection of doubles with set dimensions (n, m) that that are immutable.	`[`
		`  -1, -1, -1;`
		`  -1, 8, -1;`
		`  -1, -1, -1`
		`]`

`image`	Struct containing 3 matrices corresponding to the red, green, blue channels of an image.	`(r_mat, g_mat, b_mat)`

3.2 Lexical Conventions

3.2.1 Identifiers

The convention for identifiers is camel case, where the first letter of the identifier is lowercase, but every subsequent first character of a word is uppercase (e.g. sampleImage)

3.2.2 Keywords

The keywords listed in the table below are reserved by the language, and cannot be used as identifiers.

Keyword	Description
`func`	Function declaration. Follows syntax `func name(type varName) -> returnType`
`->`	Denotes return type of function
`return`	Ends the current function execution and returns a value
`void`	Indicates that function has no return value
`true`	Boolean keyword for true
`false`	Boolean keyword for false
`int`	32-bit signed integer
`double`	64-bit floating point number
`bool`	8-bit boolean
`string`	Array of ASCII characters
`matrix`	Mutable data structure storing any size of primitive data types
`for`	Standard for loop that executes statements when a condition related to a variable that gets incremented or decremented is true. Var must be initialized before being used in the loop. Follows syntax `for(var; condition; incr) { statements }`
`while`	Standard while loop that executes statements when a condition holds true. Follows syntax `while(condition) { statements }`
`if`	Standard if, else condition clause. Follows syntax `if (condition) { statements } else { statements }`
`else`	

3.2.3 Literals

Integer Literals
A sequence of one or more numerical digits representing an integer. Ex: `[0-9]+`

Double Literals
A sequence of zero or more numerical digits followed by a `'.'`, followed by one or more numerical digits. Ex: `[0-9]*'.'[0-9]+`

Boolean Literals
Consists of two keywords `true` or `false`.

String Literals
A sequence of character primitives enclosed by a pair of double quotation marks representing an unnamed string. Ex: `"this is a string"`

Matrix Literals
A sequence of doubles enclosed by square brackets, with rows delimited by semicolons, and items delimited by commas. Ex: `[ -1.0,-1.0,-1.0; -1.0,8.0,1.0; 1.0,1.0,1.0 ]`

3.2.4 Comments

Single-line Comments
Single-line comments are denoted by `//`. Ex: `// this is a single-line comment`

Multi-line Comments
Multi-line comments are denoted by `/* */`. Nested multi-line comments are not supported. Ex:

```
/* this is a
   multi-line comment */
```

3.2.5 Scalar Operators

Operator	Description
+	Addition (binary operator between two ints or two doubles). Will throw error if called on an int and double.
-	Subtraction (binary operator between two ints or two doubles). Will throw error if called on an int and double.
*	Multiplication (binary operator between two ints or two doubles). Will throw error if called on an int and double.
/	Division (binary operator between two ints or two doubles). Will throw error if called on an int and double.
%	Modulo (binary operator between two ints). Will throw error if called on an int and double or double and double.
<, >, <=, >=	Greater than, less than, greater than or equal to, less than or equal to, equal to, not equal to. Binary operator between two ints or two doubles. Will throw error if called on an int and double.

3.2.6 Matrix Operators

Operator	Description	Example

+	Addition (binary operator between two matrices or a scalar and matrix). Will throw error if matrix dimensions are incompatible.	`3.0 + [ -1.0,-1.0,-1.0; -1.0,8.0,-1.0 ]`
		`= [ 2.0,2.0,2.0; 2.0,11.0,2.0 ]`
		
		`[ 1.0,1.0 ] + [ 2.0,2.0 ]`
		`= [ 3.0,3.0 ]`

-	Subtraction (binary operator between two matrices or a scalar and matrix). Will throw error if matrix dimensions are incompatible.	`[ -1.0,-1.0,-1.0; -1.0,8.0,-1.0 ] - 3.0`
		`= [ -4.0,-4.0,-4.0; -4.0,5.0,-4.0 ]`
		
		`[ 1.0,1.0 ] - [ 2.0,2.0 ]`
		`= [ -1.0,-1.0 ]`

*	Multiplication (binary operator between two matrices or a scalar and matrix). Will throw error if matrix dimensions are incompatible.	`3.0 * [ -1.0,-1.0,-1.0; -1.0,8.0,-1.0 ]`
		`= [ -3.0,-3.0,-3.0; -3.0,24.0,-3.0 ]`
		
		`[ 1.0,2.0,3.0; 2.0,3.0,4.0 ] *`
		`[ 1.0,2.0; 3.0,4.0; 5.0,6.0 ]`
		`= [ 22.0,28.0; 31.0,40.0 ]`

[r, c]	Matrix access.	`matrix m =`
		`  [`
		`  -1.0, -1.0, -1.0;`
		`  -1.0,  8.0, -1.0;`
		`  -1.0, -1.0, -1.0`
		`  ];`
		`print(m[1, 2]) // should print element in 2nd row, 3rd column: -1.0`

3.2.7 Image Properties

For the code examples, assume that an image with the variable name `im` has already been declared and populated with matrices.

Field	Description	Example
`im.red`	Accesses the first matrix in an image struct. This matrix corresponds to the image's red color channel.	`matrix red = im.red;`
`im.green`	Accesses the second matrix in an image struct. This matrix corresponds to the image's green color channel.	`matrix green = im.green;`
`im.red`	Accesses the third matrix in an image struct. This matrix corresponds to the image's blue color channel.	`matrix blue = im.blue;`

3.3 Syntax Notations

3.3.1 Expressions

Precedence and Associativity Rules

Precedence	Operator	Token	Associativity
1	Parenthetical Grouping	()	Left
2	Matrix Access Function Call	[] () ->	Left
3	Unary Operator	!	Right
4	Binary Multiplicative Operators	* / %	Left
5	Binary Additive Operators	+ -	Left
6	Comparative Operators	< <= => >	Left
7	Equality	== !=	Left
8	Assignment	=	Left
9	Sequencing	;	Left

Type Conversions
The user must explicitly cast between `doubles` and `ints`. Ex: `int x = (int) 5.0;`

Equality
The equality operators (`==` and `!=`) are structural for primitive types, meaning that they recursively compare the values of primitives. However, all other types use referential equality.

Subscripts
A postfix expression in square brackets is a subscript, whose expression has type matrix.

3.3.2 Declaration

Matrix Declaration
`matrix name = [a, b, c; e, f, g; h, i, j];`
Ex: `matrix m = [ 1.0,2.0,3.0; 4.0,5.0,6.0; 7.0,8.0,9.0 ];`

The matrix specifier defines the variable as a matrix type. The elements within the matrix must be doubles. Semicolons separate every row, while commas separate elements within each row.

Image Declaration
`image name = (matrix name, matrix name, matrix name);`
Ex: `image img = (m, m, m);`

The image specifier defines the variable as an image type. The elements within the image must be matrices that have already been defined, i.e. have a variable name associated with them. The first matrix represents the red color channel, the second matrix represents the green color channel, and the third represents the blue color channel. Commas separate the matrices.

Function Declaration
`func name(type varName) -> returnType { }`
Ex: `func main() -> int { return 0; }`

Use the keyword `func` to declare a function declaration. After the function declaration, the user specifies a function name, followed by parentheses. In the parentheses, the arguments (type and name of argument) that the function accepts is defined. After the parentheses, an arrow represented by `->` followed by a data type signifies the return type of the function.

3.4 Standard Library Functions

3.4.1 Functions

Name	Description	Return Type
print(_____)	Prints argument to standard output. Can take in a string, double, float, boolean, or matrix.	void
dim(string s)	Gets the dimensions of an image file and stores those dimensions in a 1 by 2 matrix where the first element is a double representing the number of rows and the second is a double representing the number of columns.	matrix
load(string s)	Loads an image into an image object, which contains three matrices corresponding to the red, green, and blue channels.	image
save(image im)	Saves an image object as a jpg image named "image_out.jpg".	void

3.4.2 Load and Save Functions

The standard library will provide methods to load an image file (.jpg) as matrices and to save matrices as an image file (.jpg). Our language's representation of an image will consist of three separate matrices per image to represent the red, green, and blue color channels. These methods will be written in C to leverage OpenCV and linked to our implementation. Specifically, our load method will use OpenCV's imread() function to read an image file into a C Mat object, a multi-channeled matrix. We can then iterate through the Mat object and parse it to create three separate RGB matrices in our language's format. Our save function will work similar, leveraging OpenCV's imwrite() function to parse our matrices into a Mat object and write it back into an image file.

3.4.3 Image Filters

Function	Description
blur(string s)	Blurs an image and loads into an image object, which contains three matrices corresponding to the RGB channels
brighten(string s)	Brightens an image and loads into an image object, which contains three matrices corresponding to the RGB channels
grayscale(string s)	Converts an image to black and white and loads into an image object, which contains three matrices corresponding to the RGB channels
edgedetect(string s)	Detects the edges of an image and loads into an image object, which contains three matrices corresponding to the RGB channels

3.5 Code Samples

3.5.1 GCD Algorithm

```
// greatest common denominator function in VSCOde
func gcd (int m, int n) -> int {
    while (m > 0) {
        int c = n % m;
        n = m;
        m = c;
    }
    return n;
}
```

3.5.2 Grayscale

```
// grayscale filter without using built-in function
func applyGrayscale (string imageName) -> void {

    // load file into an image type
    image img = load(imageName);

    // apply filters to individual color channel matrices
    matrix newR = img.red * 0.3;
    matrix newG = img.green * 0.59;
    matrix newB = img.blue * 0.11;

    // create a new image and save it
    image newImg = (newR, newG, newB);
    save(newImg);
}
```

3.5.3 Edge Detection

```
// use built-in function for edge detection
func main() -> int {
    image im = edgedetect("test.jpg");
    save(im);
    return 0;
}
```

4 Project Plan

4.1 Process

4.1.1 Planning and Meetings

Team meetings for VSCOde ranged from quick update meetings to long hours of coding together, but all were to ensure that we were on track to complete our language.

We set up meetings twice a week, on Monday mornings and Wednesday nights. Our Monday morning meetings typically revolved around updates, quick code fixes, and delegating tasks. At the very beginning of the semester, our Wednesday night meetings largely revolved around researching. Once we started programming more heavily, we realized that we were much more efficient when pair programming or doing individual work. Towards the end of October and during December, when our code became much more modular, we set up meetings multiple times a week between various pairs or trios based on availability and subject matter.

We would also meet with our TA mentor, Mark Mazel, every Tuesday night. While meeting with Mark, we would provide updates, ask prepared questions, and note any feedback regarding the design and implementation of our language.

Outside of these in-person meetings, the team would communicate amongst ourselves daily through Messenger, particularly about design decisions made while pair programming and about roadblocks. We would also communicate with Mark through e-mail when we had urgent questions.

4.1.2 Development

In our language's nascent stages, we made decisions about the high-level design of the language together, such as those regarding syntax, basic types, operations, etc. We would read through MicroC's code before starting to code each file (ast, parser, scanner, semant, codegen), meeting once to go through questions.

After the Hello World deadline, we split up work by creating Github issues and assigning each issue to one or more members of the team to complete. What became our biggest implementation challenges are highlighted in the Project Timeline section. During the execution of these tasks, we did not program as a whole team and often split up into pairs or worked individually. Thus, these pairs or individuals would often make decisions regarding language design, based on what they experienced to be the most intuitive to understand and implement, then inform the rest of the team through Messenger.

Our delegation of tasks and code development were largely centered around types (specifically matrices and images), operations using those types, and image features. Because we were in constant communication, each team member knew about the progress being made and could help with or pick up development if another member was busy.

4.2 Style Guide

While we did not have any set guidelines during development, we inherently followed these general guidelines:

OCaml:

- The "`in`" in a "`let in`" statement goes on new line
- Tabs are 2 spaces
- `if ... then` on one line

- `else` on a new line
- No maximum on characters per line, but each line should express a complete "thought"
- If something spills over into new line, enter then 4 spaces
- If defining multiple variables in a `let` statement, the `and`s go on new lines
- Arrows `->` for pattern matching are always left aligned
- If you are within a pattern matching block and you need to go into a new line, then make sure to tab 3 times
- Descriptive, snake case variable names
  - Snake case variable names excludes AST types, which were camel case
- Group functions and definitions together and indicate them through comments
- Comment and label logic that may be unclear

C++:

- Opening curly brackets when defining functions do not begin on a new line
- Tabs are 4 spaces
- Variable names are camel case
- Two enters between functions
- The `else` starts on the same line as the ending curly bracket `}` of the corresponding `if` statement

4.3 Project Timeline

Date	Milestone
September 19	Project Proposal due
October 15	Language Reference Manual due
October 15	Parser due
October 15	Scanner due
November 14	Hello World due
December 1	Matrices Implemented in Codegen
December 3	Matrix Binop implemented in Codegen
December 5	Images Implemented in Codegen
December 7	Link in Simple C++ Tests
December 14	Load with OpenCV
December 15	Save with OpenCV
December 15	Built-In Filters Finalized
December 19	Presentation
December 19	Final Report due

4.4 Team Roles

Member	Role
Jessica Cheng	Team Manager
Hana Mizuta	System Architect
Anna Lu	System Architect
Spencer Yen	Language Guru
Kenny Yuan	Tester

4.5 Software Development Tools

We used the following languages and evironments to complete VSCOde:

- **Languages:** OCaml 4.07, LLVM 5.0, C++, bash
- **Compiling and Testing:** VirtualBox 5.2.22
- **Programming Editor:** Sublime, Atom, Vim
- **Version Control:** Github
- **Operating Systems:** OSX 10.14
- **Documentation and Presentation:** Overleaf, Google Docs, Google Slides
- **Communication:** Facebook Messenger

4.6 Project Log

Our Github commit history can be found in the Appendix.

5 Language Evolution

At a high level, the core functionality that we needed to implement in order to manipulate images is matrices. Our design and implementation of this core functionality changed as we started writing semantic checking, and later codegen. The main use case we will consider is a function to load an image file, which we planned to link in the OpenCV C++ library.

Our initial idea for load() was to do multilocal assignment (for example: matrix [100][100] r, g, b = load("image.jpg")) and load the red, green, and blue matrices directly into matrices r, g, b. When we were implementing our semantic checking, we decided it would make more sense to just wrap these matrices in an Image type that contains the r, g, b matrices. This design change would enable users to easily pass around whole images (with all r g b channels), instead of dealing with each image as three separate matrices to keep track of.

We also initially thought that we needed a matrix's dimension declared in the type in order to allocate the appropriate amount of space for it. While implementing semantic checking, we decided to introduce two matrix types, Matrix and DimMatrix(r,c), to allow the user to only interface with a single "Matrix" type without predetermined dimensions. Our intention was to allow matrices to be more flexible to the user, since the user wouldn't need to know the matrix dimensions when they declare it. We'd previously thought this wasn't possible since we need to know the dimension of the matrix to allocate the space, however, we realized that we could find a matrix's intended dimension from the expr that the matrix variable is being assigned to.

To do this, we basically converted any instance of Matrix (no dimension) into a DimMatrix(r,c) (with dimensions on the AST type) once the matrix was assigned to a value that we know the dimension of. At this point, we take the dimensions of the value and allocate the corresponding amount of space for the matrix. For example, if a matrix is being assigned to a MatLit, we can find the dimension of the MatLit first, and then allocate that corresponding amount of space for the matrix to store the MatLit. Another example is doing a matrix binop: we can find the dimension of the intended result of the matrices being operated on (adding two matrices = new matrix of same size, multiplying two matrices = new matrix of size matrix 1 row * matrix 2 col). In codegen, we kept track of any local matrix variable's dimension in a StringMap called mat_dim_map, which we referenced any time the local matrix was used. While this solution seemed initially attractive in the simplicity it granted the user, it only works at compile time because we can't resolve the runtime dimension at runtime. The biggest implication is that we couldn't return matrices in

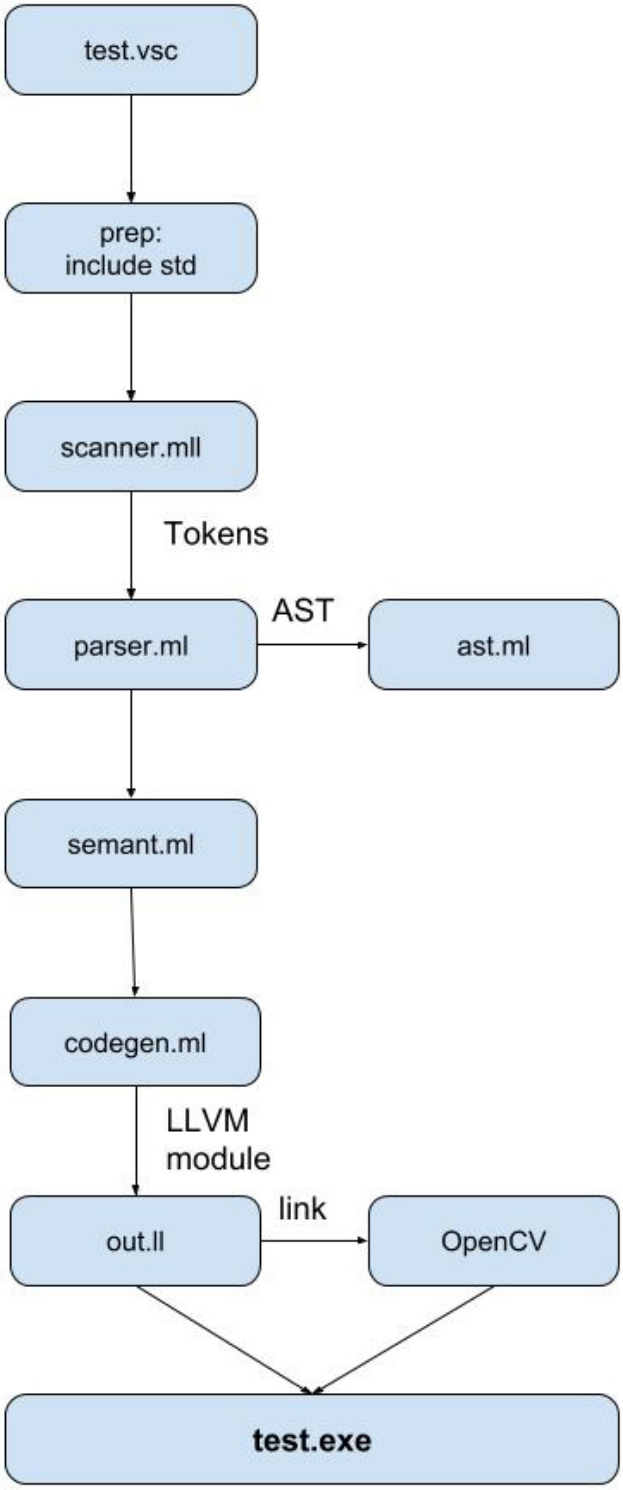
functions. Unfortunately, we only realized this late into the project after we had finished semant, and all matrix operations in codegen. We decided to move forward with this technical debt since it didn't prevent us from implementing the scope and functionality of image manipulation, and due to limited time constraints.

Since we couldn't resolve the size of a matrix at runtime, we had to make a decision to enforce a constant image size to implement our load() function. While we could derive the image dimensions if it was assigned to an image literal (3 matrix literals), there was no way we could derive the image dimensions if it was assigned to the return value of a function (load). The return value of our built in load function would then return an Image type that is a struct of 3 2D double arrays of a constant size (r, g, b matrix). The size of these images was set at the top of our codegen and utils.cpp, as constants image_row_size and image_col_size.

The final limitations of our language resulted from that fact that we implemented DimMatrix as an LLVM 2D double array, and always allocated it on the stack.

Note: After our presentation with Professor Edwards, he advised us to write an LLVM FOR loop to optimize the IR created, which was a few hundred thousand lines of build_gep, build_load, and build_store instructions for each entry in the matrix. He also advised us to move our matrix binop operations, and more of our image load and save logic to C++ functions instead of trying to do all of it in OCaml (C++ compiler is optimized). We spent the rest of the day trying this. In order to move more of the logic in load() into C++, we decided to have our C++ load function return a 3D double array (double ***: [red_mat, green_mat, blue_mat]). We were able to get the structure of passing the loaded r, g, b matrices into LLVM, and we were successful in optimizing the generated IR from 100,000+ lines to 60 lines. However, at run time we found that the values in the loaded image were non-deterministic garbage values – occasionally it found the correct values, but the majority of the time we would get random float values. We suspect this is because the memory of the C++ 3D double array is getting overridden/deallocated, making the pointers point to garbage values.

6 Translator Architecture



6.1 vscode.ml

This is the top-level of VSCOde compiler, where we invoke the prep, scanner, parser, semant, and codegen modules to generate the LLVM IR, and dump the module.

6.2 test.vsc

This is a sample VSCOde program that needs to be compiled.

6.3 prep.ml

Include any standard libraries.

6.4 scanner.mll

The scanner reads the VSCOde code and performs lexical analysis by tokenizing the input. It is passed onto the parser if it passes or rejected at the scanner level if there is an illegal character.

6.5 parser.mly

Read tokens from scanner module, making sure they are syntactically correct. If parsing doesn't generate an error , it will generate the abstract syntax tree(AST).

6.6 ast

The abstract syntax tree representation that contains the semantic structure of each of our calls. Note that we didn't implement an SAST, so semant simply accepts or rejects without modifying the types.

6.7 semant.ml

This ensures the AST is semantically correct and passes the AST representation to the codegen module.

6.8 codegen.ml

This contains the code for matrix memory storage and generates the LLVM intermediate representation instructions used to perform matrix operations.

6.9 OpenCV

OpenCV is linked with the LLVM bytecode and the helper functions in utils.cpp that perform load, save, edgedetect and other filter operations.

7 Test Plan and Scripts

7.1 Development Cycle

Features in our language were developed independently on separate branches (when possible), and we tried as much as we could to write much of our test cases before development to clearly lay out the expectation for what should and should not work. Obviously, this led to much tweaking of our tests, as we frequently ran into roadblocks and had to reassess the ways in which we carried out our implementations. But it held us accountable for making sure that our final product was close in functionality to our original intentions.

7.2 Test Script

After each feature passed its own specific tests, we ran our entire test suite to ensure that it was well-integrated with the overall system. We often tested individual test (.vsc) files using our `./link.sh` script, while our `./testall.sh` script runs our entire test suite consisting of normal (test-*) and fail (fail-*) tests. Each fail test has a corresponding output file (.err) with the expected error message, while each normal test has a corresponding output file (.out) with the expected print results. In total, we had 94 tests (53 normal tests and 41 failing tests) that cover a wide array of features in our language from standard logic and loops to linked C++ functions.

Our `./testall.sh` script compiles all tests in the tests/ folder by calling our `./link.sh` script and compares the results of each test with the corresponding output file.

7.3 Test Code

Both the `./testall.sh` script and our full test suite are available in our Appendix.

8 Lessons Learned

8.1 Jessica Cheng

I won't lie, this class was incredibly challenging. It's hard to structure and hard to teach, especially because of the size of the project. You need to learn to read code in a language you are not yet familiar with, as some of the stages of the project are due before we get a chance to really review them during lecture. A lot of early design decisions are based off of previous projects' code, so make sure you understand the pros and cons of how they implemented certain features. The TAs are incredibly helpful in talking you through ideas (shout out to Mark!), but it's still imperative that your team has an in-depth understanding of everything you're doing and looking at.

Developing new technical skills aside, the project is also challenging in the fact that it's team-based. The five of us were friends beforehand, so discussions, negotiations, and conflict resolution about code or timing was not as big of an issue as it probably would have been if we had been strangers. However, there were still issues of efficiency (especially in the beginning of the project).

My three biggest pieces of advice are:

- Try to make sure that *everyone* on your team is free for long nights from the weekend after Thanksgiving break to this project's due date because that's when a lot of the work picks up
- Don't make design decisions because they are easier to understand—think through the consequences of each decision and don't forget about the big picture
- Figure out how your team programs most efficiently and stick to it (for us, it was pair programming).

You'll run into problems you didn't foresee and Murphy's law (anything that can go wrong will go wrong) will 100% kick in at some point, whether it be with your code or with something else, so be sure budget your time as best as you can and GET SOME SLEEP! PLT was a rewarding class to take and I am of the opinion that it is a great simulation of completing a large, programming-based team project in the real world.

8.2 Anna Lu

TLDR: Add a level of indirection, and consult your TA continuously.

Long version: This class was really fun to take and I've definitely learned a lot about language design, implementation, and how to work on projects with large teams. It's honestly a must take because it ties together knowledge from Fundies, CS Theory, and AP. And it's exciting to them all come together through a project that you build yourself.

Everyone says to start early, but I would also add that even/especially if you start early, a) make sure that you're always checking your design decisions with your TA because you won't fully understand each aspect of the pipeline, and b) still expect to be changing pretty drastic aspects of language during dead week.

Our group worked really hard throughout the semester to stay on top of deadlines and hold each other accountable for our work. We had recurring meetings 2-3 times a week to check in and were meticulous about assigning Github issues and testing incrementally, but still ran into a few issue during dead week when our IR took longer than we expected to compile. At that point, we didn't have too much time/energy to scrap everything we had done, so I would recommend having a relatively light finals schedule where you can crank out the project towards the end when your understanding of LLVM and codegen is the most robust (by taking less classes or more classes with papers). Overall, it was an interesting course that I would definitely recommend! You learn a lot regardless of whether your project ends up being spectacular.

8.3 Hana Mizuta

This four month project was by far the most challenging CS project I have done, and the first serious CS project I have worked on with others. I enjoyed spending so much time on just one project because it forced me to get very familiar with OCaml and the layout of the code overall. I specifically have two big takeaways from this project. First, the project taught me how to read code. We referred to previous projects files especially when implementing matrices and linking in the C++, which taught me how to read code faster. In the future, I hope I will spend a few extra minutes when reading through the files to not only understand the one function I am extracting, but the layout of the code on a higher level - the problem of allocating matrices on the stack vs heap was definitely one that could have been avoided if we had just read previous project codes more carefully. Second, this project also taught me the importance of keeping scalability in mind. While our stack-allocated matrices were perfect for small (2x2, 3x3) matrices, they were far too clunky to handle anything over 100 pixels x 100 pixels (we originally were hoping for 4k images...). We forgot to keep the final goal of the project in mind when we were slowly testing and working through our project, and simply kept charging ahead, ignoring all of the red flags when we were trying to finish allocating space for our project. Finally, this project also taught me the importance of keeping track of the work that we each did - we originally started the project writing everything down on a Google Doc, but moving to GitHub issues helped us a lot. I wish we had written everything that we wrote in our LRM in our issues so that we could have kept track of our progress better. I learned more in this one CS class than I have in any CS class previously, and am very glad that I took PLT!

8.4 Spencer Yen

This was a challenging and large scale project that really picked up in difficulty in the final stages, when poor early design decisions affected our implementation of functionality (built in functions load and save) that we thought would be simple to implement. Aside from the language design decisions that I discuss in detail in the language evolution section of this report, my biggest learning from this project was how to work on a large software project with a team. Even with version control, a task management system, and consistent meeting times for our team and with our manager (TA), there's a lot more factors that contribute to a successful team. For our group with 5 members, I found that pair programming was the most effective, especially when different pairs alternated with completing tasks/features. This forced us to cross check design decisions and clearly articulate what we were doing, albeit sometimes at the cost of making significant design decisions without "approval" of the rest of the team or our TA. Also, I learned the importance of test-driven development to cover all possibilities of what can happen. We implemented features and usually tested with a simple use case, but instead should have more rigorously tested throughout. Always play the game: how can I break this feature?

While common advice to "start early" is generally good, starting too early before you fundamentally understand what you're implementing and why you're doing it one way can be equally dangerous. This class's material is tricky to teach in that there's a lot of concepts and moving parts that you have to understand before you really understand the big picture – only towards the end of the project did I fully grasp how all the moving parts worked together. We didn't have the foresight to catch a poor design decision, but closer review with someone who knows the full structure of how the compiler/LLVM works can. My advice would be to go over the AST structure and types in codegen with your TA/Professor in depth before implementing the rest of codegen.

8.5 Kenny Yuan

Designing and implementing your own language and compiler is not a trivial task, and I think at many points we did not put enough thought behind the methods we chose. There is a large selection of resources at your disposal, and it's important to try to use anything you can get a hold of from old code samples to the TAs. In doing so, however, it's important to make sure you understand why certain design choices are being made at every step of the way, because if you blindly code and things appear to work, it could very well come back to haunt you later on in the development process. From a holistic standpoint, this project can be overwhelming so it's important to be proactive and set internal deadlines for every task. It's never too early to start working on something, and you'll almost certainly run out of time in one way or another.

9 Appendix

9.1 Project Log

```
commit 3dd770a86ea8b18ba7ab68ecf60baeaf2a41efae
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Wed Dec 19 23:24:03 2018 -0500

    Updated test-all timeout to accomodate slow save and image load functions

commit 49ece91cd5889211d5aced16162cc528e1528937
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Wed Dec 19 23:21:20 2018 -0500

    Changed image size back to 50, reimplemented semantic checking for imagelit

commit 1a5d224c055c484accdf78364c815a541f5de6e1
Merge: 97db98b c7c79cb
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Wed Dec 19 22:07:31 2018 -0500

    Merge branch 'master' of https://github.com/mizutahana/VSCOde

commit 97db98b1c5bdaedf3e6fe9ce3ca1e7291017c997
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Wed Dec 19 22:06:28 2018 -0500

    Updated test output files, removed test_cpp and specialcpp

commit c7c79cbe49824e0260807102b0ff143d46b20a89
Merge: 7d2e3a4 141594e
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Wed Dec 19 21:47:53 2018 -0500

    Merge branch 'master' of https://github.com/mizutahana/VSCOde

commit 7d2e3a4d4d06248578ba214469786d32568e15df
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Wed Dec 19 21:47:47 2018 -0500

    mod works for binop between ints

commit 141594ec12c472c21e9e561fe31e3ceaf5ae2c7d
Merge: c735783 4f459d6
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Wed Dec 19 21:38:09 2018 -0500

    Merge branch 'master' of https://github.com/mizutahana/VSCOde

commit c7357835d44f437592772e666d0a13ba48eb2e6d
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Wed Dec 19 21:38:07 2018 -0500

    Added matrix/image return semantic checking, cast checking, fixed utils

commit 4f459d650e449124f12aac61f29f885f42a5111b
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Wed Dec 19 21:29:23 2018 -0500

    Removed string concat
```

```
commit fb5361005f91574499d4d6d2aa51d86fadb073af
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Wed Dec 19 21:24:40 2018 -0500

    Fixed broken utils.cpp

commit e80ab79dbc19bfe301803e031ccd6ad28b0f2298
Merge: c144200 355f6ba
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Wed Dec 19 21:21:14 2018 -0500

    Merge branch 'master' of https://github.com/mizutahana/VSCOde

commit c144200a3d796e897d68e291f2812e456a46efe2
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Wed Dec 19 21:20:14 2018 -0500

    Cleaned up codegen fully

commit 355f6babfd9d329653db652eadfd7a2a05c48d99
Merge: 2f5d70f 0a64d4b
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Wed Dec 19 21:06:56 2018 -0500

    Merge branch 'master' of https://github.com/mizutahana/VSCOde

commit 2f5d70f03e4d0ec05a830085c4034398b2495738
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Wed Dec 19 21:05:35 2018 -0500

    Merge branch 'master' of https://github.com/mizutahana/VSCOde

commit 0a64d4b44293bb69de32e483a667f08e4d23a66b
Merge: 0d78fe3 fb4c97e
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Wed Dec 19 21:04:01 2018 -0500

    Merge branch 'master' of https://github.com/mizutahana/VSCOde

commit 0d78fe3a1ada684dbb641cbf0a5b0ba658243cc3
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Wed Dec 19 21:03:55 2018 -0500

    Cleaned up codegen

commit fb4c97e46f572646a50981ae8d3c5aa494213c44
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Wed Dec 19 19:52:03 2018 -0500

    Cleaned up ast.ml

commit f31e347817d980a8680acd18a1599361a1ed9ede
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Wed Dec 19 19:50:26 2018 -0500

    Cleaned up compile, renamed it link.sh

commit d6e1a57fab981b30a21f36662e9174fbff7fc74a
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Wed Dec 19 19:47:56 2018 -0500
```

```
    Cleaned up parser.mly

commit 7549fa8e917cf18666cbbb36ba068aa61710c6ff
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Wed Dec 19 19:46:04 2018 -0500

    Cleaned up scanner.ml

commit 94d61e35b6c9fa6258cca52a5e84e59c026013c1
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Wed Dec 19 19:45:17 2018 -0500

    Cleaned up semant

commit 651c889671b54bec8fcb5b1daecb3e77983a825e
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Wed Dec 19 19:19:13 2018 -0500

    Cleaned up testall.sh

commit 7c238cf094d6ade1437e8d6eef5c1f022882865a
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Wed Dec 19 19:17:13 2018 -0500

    Cleaned utils.cpp

commit 28aa99395db9644bab18ae78544e241a6f2a6146
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Wed Dec 19 19:11:21 2018 -0500

    Cleaned vscode.ml

commit a4e9419d971bce275f8fb735225cb8d32a281c61
Merge: ac6a753 45b326d
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Wed Dec 19 18:45:52 2018 -0500

    Merge branch 'master' of https://github.com/mizutahana/VSCOde

commit 45b326d1bec96ff0d4f785400046c87fb0160eba
Merge: 2a1fc3b e3d4e41
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Wed Dec 19 12:42:34 2018 -0500

    Merge branch 'master' of https://github.com/mizutahana/VSCOde

commit ac6a75359f58d97b43cdb8c546b1877e543eed25
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Wed Dec 19 12:40:39 2018 -0500

    Changed load within codegen to take as pointer. Still work in progress

commit e3d4e41003b936abc6203d64978f19ed41f2a3f5
Merge: 6409712 4068296
Author: annalu <anna.lu@columbia.edu>
Date:   Wed Dec 19 12:14:37 2018 -0500

    Merge branch 'master' of https://github.com/mizutahana/VSCOde

commit 406829689626291bf78cfd4e04d3da422138b6a8
```

```
Merge: 1d14d30 8051bbc
Author: Anna Lu <anna.lu@columbia.edu>
Date:   Wed Dec 19 12:13:53 2018 -0500

    Merge branch 'master' of https://github.com/mizutahana/VSCOde

commit 1d14d30811ffea9b5701e286ad2d31831045e03b
Author: Anna Lu <anna.lu@columbia.edu>
Date:   Wed Dec 19 12:13:19 2018 -0500

    minified test image

commit 6409712b3c5e3c7038afc0883dbd1b5a0802270f
Merge: c8c4838 8051bbc
Author: annalu <anna.lu@columbia.edu>
Date:   Wed Dec 19 12:11:23 2018 -0500

    resolved merge conflict

commit c8c483861d8e91387b7cd4660a7d0494859c9d35
Author: annalu <anna.lu@columbia.edu>
Date:   Wed Dec 19 12:06:37 2018 -0500

    fixed codegen a bit

commit 8051bbcf5ba3753da65efffe092f9eb0566861b6
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Wed Dec 19 12:05:25 2018 -0500

    Making load into a for loop via cpp. Partial implementation

commit 93123a98de17d31e120d03dfdbf9f41af385f8a0
Author: Anna Lu <anna.lu@columbia.edu>
Date:   Wed Dec 19 11:41:07 2018 -0500

    debugged

commit 66a5e0ee441bb9d1353bccafb9b0f3c6e184600c
Author: Anna Lu <anna.lu@columbia.edu>
Date:   Wed Dec 19 11:32:47 2018 -0500

    added mini test image

commit b2734f2ccfb0dbfd2f6566305a4edd9a971634bb
Author: Anna Lu <anna.lu@columbia.edu>
Date:   Wed Dec 19 11:27:32 2018 -0500

    rewrote load to less lines of llvm. Pushing to test within VM

commit 2a1fc3bb9d837886baa8bd619e980cb24def48d4
Merge: d57c9ed 0ceb6a7
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Wed Dec 19 10:52:49 2018 -0500

    Merge branch 'master' of https://github.com/mizutahana/VSCOde

commit d57c9ed6f864a1b812a8702ad9f9fee26906e924
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Wed Dec 19 10:52:28 2018 -0500

    New images
```

```
commit 0ceb6a700fec8cd2f91a5cf89751fb0bc96c49d3
Author: kennyyuan98 <kenny.k.yuan@gmail.com>
Date:   Wed Dec 19 02:24:51 2018 -0500

    Tweaked demo code.

commit cd805ed254dc92efecb04fcd4c996c133bfaebfa
Author: kennyyuan98 <kenny.k.yuan@gmail.com>
Date:   Tue Dec 18 23:49:19 2018 -0500

    Changed image size to 50x50.

commit aea56bd28f462574bc9ce03734e935cd23cd8349
Merge: ecbc28a 2e454c2
Author: kennyyuan98 <kenny.k.yuan@gmail.com>
Date:   Tue Dec 18 23:41:00 2018 -0500

    Merge branch 'master' of https://github.com/mizutahana/VSCOde

commit ecbc28a445d37aff5db151cf637ed389d7c5c24b
Author: kennyyuan98 <kenny.k.yuan@gmail.com>
Date:   Tue Dec 18 23:40:51 2018 -0500

    Wrote demo code.

commit 2e454c228a84a4087eb8adaf86e916edcdaa3798
Merge: 26109af 33eb24d
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Tue Dec 18 22:58:41 2018 -0500

    Merge branch 'master' of https://github.com/mizutahana/VSCOde

commit 26109af4d8a1221ecd2db19708dfdc33ddae6ab1
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Tue Dec 18 22:58:31 2018 -0500

    edge detected photos

commit 33eb24d7e96e80021198df488cde79b501c773ea
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Tue Dec 18 22:51:48 2018 -0500

    adding images to test

commit cf67fce2155871669eb321f0e29c056ab72dce2a
Author: kennyyuan98 <kenny.k.yuan@gmail.com>
Date:   Sat Dec 15 23:38:38 2018 -0500

    Added a 'special' filter :)

commit 73aa0c490d61f53a766b8b8a0d4711a0bd01b0bc
Author: kennyyuan98 <kenny.k.yuan@gmail.com>
Date:   Sat Dec 15 22:56:01 2018 -0500

    Removed excess files (that I accidentally pushed earlier oops).

commit 9aac6c090a56c0b481dc6a7b55111b1196e885bd
Author: Kenny Yuan <kenny.k.yuan@gmail.com>
Date:   Sat Dec 15 22:50:36 2018 -0500
```

```
    Important image file

commit 7584af09b0e1e6a58c3bc00f718546c34b469f8e
Author: kennyyuan98 <kenny.k.yuan@gmail.com>
Date:   Sat Dec 15 22:48:59 2018 -0500

    Added tests for new filters.

commit 3a4fe0a3a7d9ceeb5931b079e2aa7207f8a815e5
Author: kennyyuan98 <kenny.k.yuan@gmail.com>
Date:   Sat Dec 15 22:27:06 2018 -0500

    Cleaned up semant and codegen, deleted extra files.

commit 059c2d51c66ec2d9cee316c15e998c32ac9ad2fd
Author: kennyyuan98 <kenny.k.yuan@gmail.com>
Date:   Sat Dec 15 22:22:19 2018 -0500

    Implemented blur, brighten, grayscale, edge detect filters.

commit b6f27c664e795dad2cf90c72b30094e50c00f709
Author: annalu <anna.lu@columbia.edu>
Date:   Sat Dec 15 20:03:16 2018 -0500

    Changed image sizes to 100

commit cbb01b73d77a2de38e603397ea5c58de6a3eddca
Merge: e1d7c85 d03bdc4
Author: Anna Lu <anna.lu@columbia.edu>
Date:   Sat Dec 15 19:10:09 2018 -0500

    Merge pull request #48 from mizutahana/anna_not_working_save

    Merged save code into master.

commit d03bdc45d18936483c726005bc11576625d56c11
Merge: b53428c e1d7c85
Author: Anna Lu <anna.lu@columbia.edu>
Date:   Sat Dec 15 19:09:39 2018 -0500

    Merge branch 'master' into anna_not_working_save

commit b53428cc03b48334fe33efb05d620c0d4c4396e6
Author: annalu <anna.lu@columbia.edu>
Date:   Sat Dec 15 18:55:13 2018 -0500

    new image file from test, pushing to access outside VM

commit ba0e42bbd9620f47f958a0527f920514564c7727
Author: annalu <anna.lu@columbia.edu>
Date:   Sat Dec 15 18:50:05 2018 -0500

    Adding image_out.jpg to access it outside the VM

commit e5c2057176a85f76bbce24e8cf296e7ad4112ae9
Author: annalu <anna.lu@columbia.edu>
Date:   Sat Dec 15 18:44:09 2018 -0500

    Save finally fully works! Hurrah!

commit 101d5f3ab5037b442cf197559352596c4ffce45e
```

```
Merge: 61c6c66 635d8a5
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Sat Dec 15 17:05:53 2018 -0500

    fixed merge conflicts

commit 61c6c66607e32803e3b400235f619ec6793cec47
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Sat Dec 15 17:02:49 2018 -0500

    new save logic, return not working

commit e1d7c853b917626b53f3efe2028cbfaf3f5de7ad
Author: kennyyuan98 <kenny.k.yuan@gmail.com>
Date:   Fri Dec 14 23:43:20 2018 -0500

    Added fail tests

commit 635d8a5e95ff75a4652fb2eb68bc383e1ba3445b
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Fri Dec 14 22:28:42 2018 -0500

    got img_struct_alloc to point to space where the image is in save

commit c78789d9e2a4f060889b5502eaa3ef466146a549
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Fri Dec 14 21:16:47 2018 -0500

    got rid of warnings that don't relate to save

    Co-authored-by: Kenny <kenny.k.yuan@gmail.com>

commit ddc4dc60baa62802d9be9ca3963a1eccf0c0360a
Author: annalu <anna.lu@columbia.edu>
Date:   Fri Dec 14 21:02:39 2018 -0500

    Save is now working

commit 1e7a8ec7344b2ac1eadeeca8a734d31ccccc766c
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Fri Dec 14 20:57:03 2018 -0500

    LOAD WORKS FUCK YEAH

commit e8917a11c9131bded1d59398c46658f663b34c95
Author: annalu <anna.lu@columbia.edu>
Date:   Fri Dec 14 18:18:07 2018 -0500

    save is working nowgit add .

commit 8ce8341094e718c2c4c2d6bc5e478a2358abcb06
Author: Kenny Yuan <kenny.k.yuan@gmail.com>
Date:   Fri Dec 14 12:06:24 2018 -0500

    Working on loading images of a fixed size (4096 x 3034)

commit ac89495a5435603ec82ea26d08853886e806a034
Author: annalu <anna.lu@columbia.edu>
Date:   Thu Dec 13 18:58:23 2018 -0500

    cool
```

```
commit c697483d678b4b4c544e7ba72e9019a34549a001
Author: annalu <anna.lu@columbia.edu>
Date:   Thu Dec 13 18:05:13 2018 -0500

    Fixed formatting

commit f1e603aa19dec880d5c0d3331ecf6cf41a9c9a7d
Author: annalu <anna.lu@columbia.edu>
Date:   Thu Dec 13 18:02:35 2018 -0500

    Flatten_mat worksgit add .

commit 80dac7a8b8de91c52926ea0a2ba7f5e93ad88b07
Author: annalu <anna.lu@columbia.edu>
Date:   Thu Dec 13 17:13:30 2018 -0500

    fixed buggit add .  thanks dean

commit d04519bacfa99ba991b958f54fddf512f0503d97
Author: annalu <anna.lu@columbia.edu>
Date:   Thu Dec 13 16:55:33 2018 -0500

    still not working, but some changes have been made

commit 47d1c6c4ff1352e33e155fb21f810242b6772d7f
Author: kennyyuan98 <kenny.k.yuan@gmail.com>
Date:   Thu Dec 13 14:59:42 2018 -0500

    Needed to get the real image dimensions from load_cpp

commit c6ae9ae64e9d28c9b1671dcf0d1b8e211a639ec4
Author: annalu <anna.lu@columbia.edu>
Date:   Wed Dec 12 23:16:25 2018 -0500

    here's the kind of working version

commit e76f6a46ef4c125ed0993a22dd8f88f4f96c4289
Author: annalu <anna.lu@columbia.edu>
Date:   Wed Dec 12 23:10:30 2018 -0500

    this is just what I have on wed night ahh

commit 090d8646f8f2720e941810ceaac0521de8f527aa
Author: kennyyuan98 <kenny.k.yuan@gmail.com>
Date:   Wed Dec 12 18:27:57 2018 -0500

    Attempting to assign image from load to an ID

commit deba436a22c418720b2326f571fe9888511b9de0
Author: kennyyuan98 <kenny.k.yuan@gmail.com>
Date:   Wed Dec 12 18:23:51 2018 -0500

    Fixed types issue in load

commit bde9dacee29f84e173d00f9d06e2d444c6267ee9
Merge: 8a6227a eeb895f
Author: kennyyuan98 <kenny.k.yuan@gmail.com>
Date:   Wed Dec 12 18:06:20 2018 -0500

    Merge branch 'anna_2' of https://github.com/mizutahana/VSCOde into anna_2
```

```
commit 8a6227a42251d995614931ced6507d64cacdd26a
Author: kennyyuan98 <kenny.k.yuan@gmail.com>
Date:   Wed Dec 12 18:06:10 2018 -0500

    Moved load function call

commit eeb895fbe16b38091a75d1a69203ee853933032c
Author: kennyyuan98 <kenny.k.yuan@gmail.com>
Date:   Wed Dec 12 16:45:36 2018 -0500

    Fixed testall to support fail tests.

commit 8f015ce9685f48f76821900f72b1000dc7d26b73
Author: kennyyuan98 <kenny.k.yuan@gmail.com>
Date:   Wed Dec 12 15:35:59 2018 -0500

    Fixing flatten_mat

commit ac082eafaf427ae4376fb41fb4edec8bc41853da
Author: annalu <anna.lu@columbia.edu>
Date:   Wed Dec 12 14:12:40 2018 -0500

    latest version of save

commit 8a6226df8f246059780bc93e551ffd9312848af1
Author: Anna Lu <anna.lu@columbia.edu>
Date:   Wed Dec 12 02:45:25 2018 -0500

    started working on save, will test in VM

commit 6328da98f2c8e64a7fdbf488f2b36eb437efb5c7
Author: Anna Lu <anna.lu@columbia.edu>
Date:   Wed Dec 12 01:45:17 2018 -0500

    messing around with save, pushing to work in VM

commit 5d6829f959e5686bd9c0a8d92a0b75a812559b1a
Author: Anna Lu <anna.lu@columbia.edu>
Date:   Wed Dec 12 01:25:45 2018 -0500

    Reformat dim and add test files

commit b85789b051a4b65c02c9c1592643e09f745a84f8
Author: annalu <anna.lu@columbia.edu>
Date:   Wed Dec 12 01:12:34 2018 -0500

    DIMENSIONS WORK!

commit c6ad5e8e2694ccf8b6726bbec34db9afa43c6225
Author: annalu <anna.lu@columbia.edu>
Date:   Wed Dec 12 00:42:53 2018 -0500

    attempt 2

commit 5e1216d613ca956a38d23f39842fb9b658672700
Author: annalu <anna.lu@columbia.edu>
Date:   Tue Dec 11 22:31:29 2018 -0500

    attempt 1
```

```
commit 720fb9fa7cb04eadfe1250e87f09e6e0ecb0c26a
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Tue Dec 11 19:11:51 2018 -0500

    Fixed our test dbl_arr func, decided to move our built in functions directly into
    call to avoid dealing with matrix pointers

    here we once again realize our mistake for implementing matrix as array array and not
    a pointer

commit 4ac41e9f78d8bff53425c3a323b2b718717ea976
Merge: b2c9fef 94822fc
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Tue Dec 11 18:33:55 2018 -0500

    Merge branch 'master' of https://github.com/mizutahana/VSCOde

commit b2c9fef02fbaf584cbd55f4beca4f6fb0570a166
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Tue Dec 11 18:30:36 2018 -0500

    Testing array cpp function

commit 94822fc596fcb9050b946da0c833a5f03bc4b401
Merge: 1fd1e7f 86dd77d
Author: kennyyuan98 <kenny.k.yuan@gmail.com>
Date:   Mon Dec 10 23:00:22 2018 -0500

    Merge branch 'master' of https://github.com/mizutahana/VSCOde

commit 1fd1e7f14b329ec9451e02ca2c77a8ee9a6a0479
Author: kennyyuan98 <kenny.k.yuan@gmail.com>
Date:   Mon Dec 10 22:58:51 2018 -0500

    Implemented getting image dimensions (almost). Currently unable to access the double*
    from cpp function.

commit 86dd77d42f42c9c9eef2ea1f62c3697cee33d424
Author: annalu <anna.lu@columbia.edu>
Date:   Mon Dec 10 22:15:56 2018 -0500

    Fixed all tests.

commit 82caa71a61762f81ddf9e445e650c3d3ad0b1a00
Author: Anna Lu <anna.lu@columbia.edu>
Date:   Mon Dec 10 21:31:01 2018 -0500

    Modified tests again

commit 1a8090ba3606b06b60a8d5856960462baa82b25a
Author: Anna Lu <anna.lu@columbia.edu>
Date:   Mon Dec 10 21:23:13 2018 -0500

    Modified tests to reflect newest syntax

commit f343f4273d84d0f618a25a5a2f2db17296a616cb
Author: annalu <anna.lu@columbia.edu>
Date:   Mon Dec 10 21:15:26 2018 -0500

    Modified tests
```

```
commit 4d39c9d9178de5ca7df41c32defe9dce3e9e1f3b
Author: Anna Lu <anna.lu@columbia.edu>
Date:   Mon Dec 10 21:07:20 2018 -0500

    Moved old tests to tests folder

commit cfdb57e95487b0c97827e89b807d02e537459fc4
Author: kennyyuan98 <kenny.k.yuan@gmail.com>
Date:   Mon Dec 10 20:37:04 2018 -0500

    Added rowsize and colsize getters for matrices.

commit 74f4d2c0af165d20f38b465b5c20da06e06c3331
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Sun Dec 9 11:09:24 2018 -0500

    Tried to update so image in mat actually loads, doesn't work yet

commit c4fc70f021b997f310e136e45d986ba069ee9aa0
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Sun Dec 9 01:21:07 2018 -0500

    Basic image load built in method

    Co-Authored-By: Hana Mizuta <hm2694@columbia.edu>

commit e8a44d00a34f0b1726a4a1fc1b3548e36fd2b69f
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Sat Dec 8 22:59:08 2018 -0500

    added image size constants

    Co-Authored-By: Hana Mizuta <hm2694@columbia.edu>

commit 4fbb29c71d2a6a0ab178c8697f1578eecef5aaad
Merge: c916df9 ed43f27
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Sat Dec 8 22:45:37 2018 -0500

    Merge branch 'master' of https://github.com/mizutahana/VSCOde

commit c916df90c237025e0dea50f1e6cf49eed898fdb0
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Sat Dec 8 22:43:17 2018 -0500

    Partial implementation of load_cpp

commit ed43f270422eb700dfe3b36c0b653c8e7192df71
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Sat Dec 8 22:38:51 2018 -0500

    Removed DimImage type

commit 98ea0cc87f7001ef581ba11691eece215666b33e
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Sat Dec 8 14:34:52 2018 -0500

    Changed testall script to link in utils.cpp, fixed matrix tests

    LIT
```

```
commit 30dfd2707377d8ba19f7b54ac2eda77d53b02478
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Sat Dec 8 12:10:20 2018 -0500

    Figured out built in function building, implemented image row size with opencv

    Co-Authored-By: Hana Mizuta <hm2694@columbia.edu>

commit 2fa45dcd9fd6219e614de44e8ef39e5540cd766e
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Sat Dec 8 03:19:04 2018 -0500

    Fixed call and changed function decl building

    Still need to figure out how to give the correct return types

    Co-Authored-By: Hana Mizuta <hm2694@columbia.edu>

commit ab9563e15d21c82922d2f6109eb11592ff5037ac
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Fri Dec 7 22:25:00 2018 -0500

    add row size to semant, print in test

commit 1ebffae1d3eb3c0e77347bd8595900f80738cd8d
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Fri Dec 7 22:15:07 2018 -0500

    row size test

commit 6ee775f3beaa146443a322d719b0d04376f69242
Merge: 872e501 7654937
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Fri Dec 7 21:02:27 2018 -0500

    Merge branch 'master' of https://github.com/mizutahana/VSCOde

commit 872e5017c4bde586281ef244e7290e4e3616895c
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Fri Dec 7 21:02:24 2018 -0500

    compile script to link in opencv, ext.cpp file

commit 7654937cb9ddf1a902cab19d46e024fa64a42f2f
Author: Anna Lu <anna.lu@columbia.edu>
Date:   Fri Dec 7 05:41:57 2018 +0000

    Fixed bugs relating to print merge

commit 624cafb4ea78a9fa4972437dfbd41e4d0c411f2d
Merge: 8b122fb f514c51
Author: Anna Lu <anna.lu@columbia.edu>
Date:   Fri Dec 7 00:24:24 2018 -0500

    Merge pull request #47 from mizutahana/universal_print

    Universal print

commit f514c5159a03264926aec98b3a10c1c8a169ade7
Merge: 244ae60 8b122fb
Author: Anna Lu <anna.lu@columbia.edu>
```

```
Date:   Fri Dec 7 00:24:11 2018 -0500

    Merge branch 'master' into universal_print

commit 244ae605520eb290b09e15900608ad885f984ad8
Author: Anna Lu <anna.lu@columbia.edu>
Date:   Fri Dec 7 05:02:37 2018 +0000

    Removed commented code for print bool attempts

commit c068199af3d53de5d14884016ae8b632c6ce74cf
Author: Anna Lu <anna.lu@columbia.edu>
Date:   Thu Dec 6 19:00:44 2018 +0000

    Implemented matrix printing for named matrices. Users may not print anonymous mat lits

commit 8b122fb2ffce1a33f65275d6fda2f935a2c933fd
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Wed Dec 5 23:45:47 2018 -0500

    Get rid of warnings

commit e202c56de0a302674eb1e46a68e3cce8f2c19686
Merge: d45b829 7e09d1f
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Wed Dec 5 23:44:21 2018 -0500

    Merge branch 'master' of github.com:mizutahana/VSCOde

commit d45b8299f2c77bfe7cc449d1497b2dabf8952504
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Wed Dec 5 23:44:03 2018 -0500

    Implemented image id assignment

commit 7e09d1f12eae462efc69da0947a5a711d8c32087
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Wed Dec 5 23:42:22 2018 -0500

    Removed warnings, I think

commit d96e1e5188bd3033aae33057236df4a48107b6ff
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Wed Dec 5 23:27:41 2018 -0500

    Edited so that it properly makes

commit dcece4c1dc300ed95e620d9f1d8384ad08836b15
Merge: 548909a 57ee3de
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Wed Dec 5 23:21:29 2018 -0500

    Merge pull request #45 from mizutahana/image_t

    image type complete

commit 57ee3deaaa34261aeb9bf38d1c9beceee85780bd
Merge: 5568970 548909a
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Wed Dec 5 23:21:13 2018 -0500
```

```
    Merge branch 'master' into image_t

commit 55689708d674f647a3436728de312dd10e8634e3
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Wed Dec 5 23:16:57 2018 -0500

    Image Access Works.. [U+FFFD][U+FFFD]

    Co-Authored-By: Spencer Yen <spencersyen@gmail.com>

commit 8710fa3b6ff7abb224294bd4ef3d919b4380d093
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Wed Dec 5 22:54:33 2018 -0500

    Matrix and Images both work. Together. A concept. Truly.

    Co-authored-by: Jessica Cheng <jjcheng1998@gmail.com>
    Co-authored-by: Spencer Yen <spencersyen@gmail.com>

commit 7cfc236c18514ddb60065a32c0432e0e1367f6be
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Wed Dec 5 22:33:47 2018 -0500

    IMAGE LITS FINALLY WORK FUCK YUUUUUUUUHHHH

commit 548909a42dd4cfd46ac1d79b3e8eb9437b0086e0
Author: Kenny Yuan <kenny.k.yuan@gmail.com>
Date:   Thu Dec 6 02:43:52 2018 +0000

    Implemented index out of bounds check on matrices.

commit 7de32471811f6f1e20c9ef3b910f448cbf9894b6
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Wed Dec 5 21:29:29 2018 -0500

    Partial updates, still not working, pushing so we can all look on indiv comps

commit e6f9eac6ab83e4915905400b5ded465872da7dda
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Wed Dec 5 17:04:58 2018 -0500

    NOT FUCKING WORKING APOLOGIES FOR THE SHIT COMMIT

commit 42eacaf13f1200a4cd63adb6e9aad09b2e6f74b6
Author: Anna Lu <anna.lu@columbia.edu>
Date:   Wed Dec 5 20:45:17 2018 +0000

    Implementing print function, working for all but bool types: currently will print 1, 0

commit 741787ca8ae6fe5bf128c2e4460f1ef325b15d49
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Wed Dec 5 13:46:54 2018 -0500

    Added image dim map

commit 7462f98be4e108c9c2d94b176e6290ed5c46c27e
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Wed Dec 5 11:26:50 2018 -0500

    Edited image_t so that image lits would be of mat lits
```

```
commit 62e54846de3a3deebeffb6cade142d0c935a7910
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Wed Dec 5 10:54:53 2018 -0500

    Added the image_t

commit 0e567bdc3eb6e931a26153ede7f1502affe2ac1b
Author: Hana Mizuta <hm2694@columbia.edu>
Date:   Wed Dec 5 10:01:58 2018 -0500

    Edited Makefile

commit 7df1729aeb3173a4d2f5e665521b86600afafa3b
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Mon Dec 3 12:45:49 2018 -0500

    Reformatted codegen.ml

commit fbcc340f0235f478f0dffde3d45c4062fe0642a9
Author: Anna Lu <anna.lu@columbia.edu>
Date:   Mon Dec 3 16:45:32 2018 +0000

    Added universal type printing to codegen. Still doesn't work in semant

commit af9c6bfbb5a229f79866f14c1558a020521c7b1f
Merge: efdb471 ae0ffd1
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Mon Dec 3 10:44:00 2018 -0500

    Merge branch 'master' of github.com:mizutahana/VSCOde

commit efdb4719f0b5eb03c2c6a60822c7eb424215e615
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Mon Dec 3 10:43:58 2018 -0500

    Fixed test all, renamed calc to vscode

commit ae0ffd19b3f98ca891a160ef3b7db8e34480643b
Merge: 3c42000 43268eb
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Mon Dec 3 00:18:23 2018 -0500

    merged raise failure on binop with unnamed matrices

commit 3c42000cebd0ea202807e6bc400a77bf4b4fe3f8
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Mon Dec 3 00:16:11 2018 -0500

    raise failure when doing binop with unnamed matrices

commit 43268ebadf2891e1e739895dff8e08e54e25c1e0
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Mon Dec 3 00:15:02 2018 -0500

    Renamed binop scalar tests

    Co-Authored-By: Jessica Cheng <jxcheng@users.noreply.github.com>

commit 4dbbf3be80fca30c0f644cc08a43d0abc0fb1c1d
Merge: d86bccc 59cf748
Author: Spencer Yen <spencersyen@gmail.com>
```

```
Date:   Mon Dec 3 00:13:11 2018 -0500

    Merge branch 'master' of github.com:mizutahana/VSCOde

commit d86bcccfba8d93b3acac7898ac86f1454e3ad64f
Merge: 97b7cce dfea68b
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Mon Dec 3 00:12:14 2018 -0500

    merge in

commit 59cf7489819b1c5c0908b9e77349031d8a6a3f99
Author: Anna Lu <anna.lu@columbia.edu>
Date:   Mon Dec 3 05:11:56 2018 +0000

    Commit binop scalar mult tests

commit 97b7cced0f6ebe4358334068e79ebeca3bf71bed
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Mon Dec 3 00:04:23 2018 -0500

    Implemented matrix multiplication between 2 IDs

    Co-Authored-By: Jessica Cheng <jxcheng@users.noreply.github.com>

commit dfea68b39a98d1ae12341bac14ce0e97ab328cdb
Author: Anna Lu <anna.lu@columbia.edu>
Date:   Mon Dec 3 04:48:22 2018 +0000

    Added scalar matrix binop operations. All working.

commit b458a0f945cb6641b911a6218a23412312edb4c4
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Sun Dec 2 21:59:50 2018 -0500

    matrix add matrix works(??) for ids but it allows you access things out of bounds

commit 83cf7fc09563cbab8166b29579aa68993ce39ff9
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Sun Dec 2 12:06:35 2018 -0500

    Made var names prettier. Found bugs

    Found bugs:
    1. Mat binop is not working

    Co-Authored-By: Kenny Yuan <kenny.k.yuan@gmail.com>

commit 40208be93816f5551d3fbab83cefef693b480e89
Author: Anna Lu <anna.lu@columbia.edu>
Date:   Sun Dec 2 05:30:13 2018 +0000

    Streamlined & improved binop documentation

commit 2ce6e638ea750604a7036752572230b8b36c2910
Author: Anna Lu <anna.lu@columbia.edu>
Date:   Sun Dec 2 04:49:17 2018 +0000

    FINALLY HAVE WORKING MATRIX ADDITION ON MATRIX LITERALSgit status

commit ddb1fe379a747da9546090486c7b48fb6f323ad3
```

```
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Sat Dec 1 23:06:23 2018 -0500

    minor updates

commit 3005d95816e95ef8f08151beca9e769ffa7445ef
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Sat Dec 1 22:24:21 2018 -0500

    matrix add matrix no longer seg faults[U+FFFD][U+FFFD]

commit b9e902c883784a14350f0b770c91f8375219e878
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Sat Dec 1 21:30:41 2018 -0500

    matrix add matrix in progress

    Co-authored-by: annalu <anna.lu@columbia.edu>

commit ea65233af49c6c1ae7857f131a3a67b76bdf2077
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Sat Dec 1 14:58:26 2018 -0500

    binop matrix op matrix addition try #1

commit afb47c6f06f42a42f431ba6219ed9c17d83ad674
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Sat Dec 1 14:33:25 2018 -0500

    binop matrix in progress

commit 10cec463e25da9c2f9b028474002e4ca1924df28
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Sat Dec 1 17:12:51 2018 +0000

    assign after noassign works

commit 87f0c57d77a926819b5310c3c098bef489cb6692
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Sat Dec 1 01:44:28 2018 -0500

    Reverted back to matrix as arrayarray, now passing around mat_dim_map to keep track
    of matrix dimensions

    Co-Authored-By: mizutahana <mizutahana@users.noreply.github.com>

commit ba805dcbc9c2a42424456f94ad88253b4a993a23
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Fri Nov 30 02:35:30 2018 -0500

    Figured out assign matlit

    Co-Authored-By: mizutahana <mizutahana@users.noreply.github.com>

commit 45a8d7fcfdce01f3325149ef0e476cb9e3ee34c0
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Fri Nov 30 02:07:23 2018 -0500

    Changed matrix to struct of (arrayarray, row, col)

    Co-Authored-By: mizutahana <mizutahana@users.noreply.github.com>
```

```
commit 1a3328458adada8cc17f5a8d7212c67aecb12bdf
Merge: 0b11521 0e2eac3
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Thu Nov 29 18:34:48 2018 -0500

    Merge branch 'master' of github.com:mizutahana/VSCOde

commit 0b11521b8eca45296b3ca47db690eed3ce25e67a
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Thu Nov 29 18:33:57 2018 -0500

    Matrix memory allocation in local (Matrix -> DimMatrix)

    Co-Authored-By: Anna Lu <annalu@users.noreply.github.com>

commit 0e2eac3fddc5a29774cc120b8c6b906c393339bd
Merge: a15d0d4 78ce720
Author: annalu <anna.lu@columbia.edu>
Date:   Thu Nov 29 21:43:43 2018 +0000

    resolved merge conflict in codegen

commit 78ce7209797f4da7213ed7ab9d13591b0cea9211
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Thu Nov 29 12:21:07 2018 -0500

    Changed MatAccess type and added Mat type

    MatAccess is now a string * expr * expr
    Need to do mat_print still. only partially implemented

    Co-authored By: @jxcheng

commit 0f23f95fb3ee6382d365754994eccdf0ed916e7d
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Wed Nov 28 21:18:52 2018 -0500

    Fixing merge error

    merge fix in commit e96ad82ac4 was wrong

commit a15d0d41098b775188608311522f4ec9b8127f36
Author: annalu <anna.lu@columbia.edu>
Date:   Mon Nov 26 05:09:39 2018 +0000

    Added unop to codegen

commit 183cd9747751c5d7b6455b3302f5c843ee56c468
Author: annalu <anna.lu@columbia.edu>
Date:   Mon Nov 26 04:51:45 2018 +0000

    Added binop to codegen

commit e45184ff064d4e91bb6fc062af917086816511d9
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Sun Nov 25 18:01:25 2018 -0500

    I think this was Kenny changing calc.ml around last Monday...?

commit e96ad82ac417ec98cce687a766b4c1ccabde5ee2
```

```
Merge: e6e4540 fcbf1f7
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Sun Nov 25 14:45:58 2018 -0600

    Fixed merge conflicts

commit fcbf1f70bb3628a863595f9ae8690608909d5793
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Sun Nov 25 14:38:56 2018 -0600

    Added print functions for double bool int. Need to add matrix later

    Co-Authored-By: mizutahana <mizutahana@users.noreply.github.com>
    Co-Authored-By: Kenny Yuan <kenny.k.yuan@gmail.com>

commit e6e4540f53bbf57db96cfc5cb824c3ea422dc924
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Fri Nov 23 20:39:32 2018 -0600

    Implemented Call in codegen

    Coauthor: @kennyyuan98 @spenciefy

commit 407e9e4ba7a4bacd920247d520a4c42318f1a598
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Fri Nov 23 20:13:23 2018 -0600

    Added casting between int <-> double in codegen

    Coauthor: @kennyyuan98

commit 250f9b7b2d7746d7b23b4813d7c3a1b682988267
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Fri Nov 23 20:06:20 2018 -0600

    add matrix and image to cast in semant

commit 892d7cbacd3eacf157dbb269a1a63b7bd4f86fd4
Author: Kenneth Yuan <kenny.k.yuan@gmail.com>
Date:   Sun Nov 18 15:19:20 2018 -0500

    Added initial tests.

commit cae73826e14fdefcb0abcde843b2b8fa316131e1
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Thu Nov 15 16:21:16 2018 -0500

    Cleaned up codegen to improve readability

commit 865b0fb67dabe9405aa8dd830027e846383cf505
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Thu Nov 15 14:25:50 2018 -0500

    COMPLETED LOCAL VAR EVERYTHING (LOCAL, ID, ASSIGN)
    WOOOOOOOOOO [U+FFFD][U+FFFD][U+FFFD][U+FFFD][U+FFFD][U+FFFD]

    Coauthors: @jxcheng

commit 144db9ced9ba397e4846bbcccbc771006e6bda12
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Thu Nov 15 13:31:06 2018 -0500
```

```
    Added Assign, but the local_var map is not properly passed around

    Coauthors: @jxcheng

commit afca086251f04f2619b30eee957713a62f548929
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Thu Nov 15 12:40:59 2018 -0500

    Added local vars to codegen.

    Coauthors: @jxcheng @spenciefy

commit a68e7ac9e743da883f4815b02cd353f5a72c468d
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Thu Nov 15 11:48:52 2018 -0500

    Reorganized codegen with expr, stmt. Added emojis.

    Co-Authored-By: Jessica Cheng <jxcheng@users.noreply.github.com>
    Co-Authored-By: spenciefy <spencersyen@gmail.com>

commit f403eba29546d7efbe958a2a44a174e855bfdb47
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Tue Nov 13 19:34:39 2018 -0500

    Removed commments in testall.sh

commit 14cb2dfe654c77b3decbb9b6a968e8689377fede
Author: Anna Lu <anna.lu@columbia.edu>
Date:   Tue Nov 13 21:51:25 2018 +0000

    Modified return type of test file to successfully build

commit 32088074902e145d1d105cf01c605370438a7a5c
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Tue Nov 13 09:41:21 2018 -0500

    Added logging to testall.sh, fails at .exe > .out (but generates .out correctly??)

commit 2692d540c2ce08613bb9936b3feb731c1e594fda
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Mon Nov 12 11:44:01 2018 -0500

    Edited calc.ml to remove warning

commit 6c4401ec98b96f24dbbb81d29d686270927f436f
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Mon Nov 12 11:43:21 2018 -0500

    Edited Makefile to clean test-hello.* file

commit 1412c811f5b257af9482f0e4af58966805dd772c
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Mon Nov 12 10:34:12 2018 -0500

    Updated testall (should have diff file)

commit 028ebd3feb69fdd30cc5b828acc9b1ac349b1b33
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Mon Nov 12 10:27:14 2018 -0500
```

```
    Revert "testall in progress"

    This reverts commit 2006ca7c5c338142ae868b5f02303f45f72999d0.

commit 2006ca7c5c338142ae868b5f02303f45f72999d0
Author: Jessica <jjcheng1998@gmail.com>
Date:   Mon Nov 12 15:21:58 2018 +0000

    testall in progress

commit 74418e851d2736e17f260e7826f6dc99ba144cf5
Author: Jessica <jjcheng1998@gmail.com>
Date:   Fri Nov 9 17:10:36 2018 +0000

    edited codegen to get rid of compilation warnings

commit 1f05e5d80883605e0c882a8e6ac05b1a7700c840
Author: Jessica <jjcheng1998@gmail.com>
Date:   Thu Nov 8 23:57:37 2018 +0000

    WE DID PRINT HELLO WORLD! still need to fix testall.sh

commit 9c7d4d308c6e623a838bf0c626cb004610e836b5
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Thu Nov 8 00:18:11 2018 -0500

    Reformated testing and played around and failed with testing.

    Sad times friends.

commit 105029f7962eebd6bd2e246f82be261f5894f73e
Merge: 1c2b9f0 de113b8
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Wed Nov 7 23:52:19 2018 -0500

    Merge branch 'master' of https://github.com/mizutahana/VSCOde

commit 1c2b9f01c8d9b8dae3fe2daade24d3e4cfa7914f
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Wed Nov 7 23:52:16 2018 -0500

    Adding testall script

commit de113b869ded55a1d818e6ffcbfb14ebad2d0c2a
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Wed Nov 7 23:49:21 2018 -0500

    Added helloworld.vsc test file, fixed calc with error catching

commit 19cb403338dcc108991ba24f1c0639636be05598
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Wed Nov 7 23:06:19 2018 -0500

    Codegen placeholders and todos, make works!

    woo hoo [U+FFFD][U+FFFD]

    Co-Authored-By: mizutahana <mizutahana@users.noreply.github.com>
    Co-Authored-By: Anna Lu <annalu@users.noreply.github.com>
    Co-Authored-By: Kenny Yuan <kenny.k.yuan@gmail.com>
```

```
    Co-Authored-By: Jessica Cheng <jxcheng@users.noreply.github.com>

commit e467e6515c7311c98b0412750230779ff732b242
Author: Anna Lu <anna.lu@columbia.edu>
Date:   Wed Nov 7 21:18:23 2018 -0500

    Removed char in semant

commit 7c91a704d9b4fbf13038a2272b46b0d3c99f1452
Author: Anna Lu <anna.lu@columbia.edu>
Date:   Wed Nov 7 21:17:11 2018 -0500

    Removed char type

commit f57acdf832bb28ee01c33e5a71717e47e126d93e
Merge: 62df576 cafabe5
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Wed Nov 7 21:03:35 2018 -0500

    Merge branch 'master' of https://github.com/mizutahana/VSCOde

commit 62df576857302cbfafce6158b391ad4e784b6e60
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Wed Nov 7 21:03:32 2018 -0500

    Edited Makefile and calc to take in codegen

commit cafabe5014eec39485b71d1613d5fa881fb1c11a
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Wed Nov 7 21:01:00 2018 -0500

    Parsing now accepts doubles in matrices

commit 0fbafa97d89436b13f204b8f8116d8b7e963310f
Author: Kenny Yuan <kenny.k.yuan@gmail.com>
Date:   Wed Nov 7 17:23:39 2018 -0500

    Fixed tests per our new matrix update.

commit a4b7ddbdd375e02e47c57cbd7ffb11fe15442c25
Merge: cb887f9 5a23768
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Wed Nov 7 17:15:11 2018 -0500

    Merge branch 'master' of https://github.com/mizutahana/VSCOde

commit cb887f96a051a7cd0168f092b4371731d51b3c5f
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Wed Nov 7 17:15:01 2018 -0500

    fixed conflict with function return type Matrix and returning a DimMatrix

commit 5a237684bcee488aa76fedffe1479f6fc7281fb0
Merge: a685f73 541c6b0
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Wed Nov 7 16:57:48 2018 -0500

    Edited test

commit a685f735fa8e33de9494fdfd81a1822b42e5d6cd
Author: Hana Mizuta <mizuta.hana@columbia.edu>
```

```
Date:   Wed Nov 7 16:56:41 2018 -0500

    Edited test

commit 541c6b01d59b1d406030221061385386efcf52e4
Author: Kenny Yuan <kenny.k.yuan@gmail.com>
Date:   Thu Nov 1 13:42:41 2018 -0400

    Added built-in functions load and save to semant.

commit 1f433d915ae5e6793d9eff718c7108b0a2705b75
Merge: 30bbf54 1ac88fa
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Thu Nov 1 11:20:14 2018 -0400

    Merge branch 'master' of github.com:mizutahana/VSCOde

commit 30bbf54a47c6301682d3e49903e4bcde69a208ac
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Wed Oct 31 22:26:46 2018 -0400

    Added semant checks for image red,green,blue functions

commit 1ac88fafecaf9cf364a953e62661de32d99b56eb
Merge: 9503d84 bdcae78
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Wed Oct 31 22:23:08 2018 -0400

    Merge branch 'master' of https://github.com/mizutahana/VSCOde

commit bdcae783ad6f4e3566277c80ebf095ec007d3216
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Wed Oct 31 22:19:28 2018 -0400

    Check if matrices in image literal are of same size

commit 2d4c3d324a8d4783707f5aebd93e1db45b168a23
Merge: 62e4858 20d262d
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Wed Oct 31 22:13:10 2018 -0400

    Merge branch 'master' of github.com:mizutahana/VSCOde

commit 62e4858cc4e954c8ee4f990e06eb7289c8bfa3f2
Merge: a1de433 0262bc1
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Wed Oct 31 22:12:48 2018 -0400

    Merge branch 'master' of github.com:mizutahana/VSCOde

    LIT DUDE

commit 20d262d2eafb34694d281edbe12bcc6903b744f9
Author: Kenny Yuan <kenny.k.yuan@gmail.com>
Date:   Wed Oct 31 22:05:24 2018 -0400

    Fixed test func1.

commit 9503d84c7e4fb892c193b764c2e3929a80223047
Merge: e18ae4d 1f0d2a9
Author: Jessica Cheng <jjcheng1998@gmail.com>
```

```
Date:   Wed Oct 31 22:04:38 2018 -0400

    Merge branch 'master' of https://github.com/mizutahana/VSCOde

commit 1f0d2a90c23e40a30b1efa129628e878e2e04be6
Author: Anna Lu <anna.lu@columbia.edu>
Date:   Wed Oct 31 22:03:46 2018 -0400

    Updated pretty printing

commit e18ae4dd46a05423b13bafc9d48f10335f890e54
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Wed Oct 31 22:03:33 2018 -0400

    updated MatAcess: stored dimensions indexed at one but accessing is 0

commit 767fc06d1afc764484828e67584574d2106e594e
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Wed Oct 31 22:02:06 2018 -0400

    updated matrix tests

commit a1de43309af6e363705f1913ee507ac8cadd203c
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Wed Oct 31 22:00:58 2018 -0400

    Added image type, new matrix without dimension and dimmat, remove multilocal and tuple

commit 0262bc1313d2b992cfdeb372328de0d1cb84a227
Author: Kenny Yuan <kenny.k.yuan@gmail.com>
Date:   Wed Oct 31 21:59:45 2018 -0400

    Fixed test for for2.

commit e888ff4aab0d211e9987f458295f20bce0432d76
Author: Kenny Yuan <kenny.k.yuan@gmail.com>
Date:   Wed Oct 31 21:56:35 2018 -0400

    Created new test for for-loops. Cannot declare int i in the for condition.

commit 18770e5be5928dcf5399c7a959d53f1cd5e5e51a
Merge: 8dcb4a4 47377d0
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Wed Oct 31 21:55:49 2018 -0400

    Merge branch 'master' of https://github.com/mizutahana/VSCOde

commit 8dcb4a459cc2dd9d1ea92466b1aadf486e3e41ab
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Wed Oct 31 21:55:45 2018 -0400

    Noassign adding to varmap works

commit 47377d00d8499cd2fcbe3d1dfcd0a42b8593faa3
Merge: f3543ad 548ac19
Author: Kenny Yuan <kenny.k.yuan@gmail.com>
Date:   Wed Oct 31 21:48:00 2018 -0400

    Merge branch 'master' of https://github.com/mizutahana/PLTEA

commit f3543ad1ecd007d25b71ea11768b55ace8696215
```

```
Author: Kenny Yuan <kenny.k.yuan@gmail.com>
Date:   Wed Oct 31 21:47:46 2018 -0400

    Fixed test for GCD. Error with printing integers.

commit 548ac19eb68e7ff44e71759e6854ed7d0b822ed5
Merge: a757b86 e3efc30
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Wed Oct 31 21:47:40 2018 -0400

    Merge pull request #7 from mizutahana/jessica

    finished MatAccess & resolved merge conflicts

commit e3efc30653c27ed7df1f8fa1bf780e21dd6c50f6
Merge: d15e2a4 a757b86
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Wed Oct 31 21:46:14 2018 -0400

    fixed merge conflicts

commit d15e2a4fe8b9662a33e97bc555291372df91cdb3
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Wed Oct 31 21:44:48 2018 -0400

    finished MatAcess

commit a757b86df42e7271951103b23120c6b62159237f
Merge: ebcc878 d0572f9
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Wed Oct 31 21:44:17 2018 -0400

    Merge branch 'master' of https://github.com/mizutahana/VSCOde

commit ebcc878675e7d6bd5aa0af405bcf1e053cd64fb7
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Wed Oct 31 21:43:59 2018 -0400

    Added casting for bool to str

commit d0572f9237ec26b6a1c47fcc23f23df4991f4818
Author: Kenny Yuan <kenny.k.yuan@gmail.com>
Date:   Wed Oct 31 21:42:50 2018 -0400

    Fixed test for add. Error was with trying to print an integer.

commit 6dbac7ee8b41b957d6e7dfa5eccda1f979a92550
Author: annalu <AnnaLu@dyn-160-39-8-153.dyn.columbia.edu>
Date:   Wed Oct 31 21:25:39 2018 -0400

    Update make clean

commit d1d47e7cbc12f2e24474212b6970c8b43d1f3308
Author: annalu <AnnaLu@dyn-160-39-8-153.dyn.columbia.edu>
Date:   Wed Oct 31 21:21:42 2018 -0400

    Added pretty printing call in calc.ml

commit 8e58e18cd61bc5d9bbbe34fc22c13c90654b2972
Merge: 11d91e6 62b6123
Author: Jessica Cheng <jjcheng1998@gmail.com>
```

```
Date:   Tue Oct 30 22:35:35 2018 -0400

    Merge pull request #6 from mizutahana/jessica

    binop for matrices (matrix & matrix as well as matrix & scalar); fixed empty matrix
    problem

commit 62b6123173afad8fdd6e01d92044c40c6a82f03d
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Tue Oct 30 22:17:26 2018 -0400

    fixed empty matrix problem

commit 3aea271a1a7636a7f9f90957deadf6d4e8cda0fc
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Tue Oct 30 21:53:00 2018 -0400

    scalar operations with matrices added. no division.

commit 3672aeabfc0c600cf8cb5bbd8f8f23c698bbc83e
Merge: 78532a4 11d91e6
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Tue Oct 30 21:37:53 2018 -0400

    matrix op matrix in binop works as expected

commit 11d91e69b2797d545e2c93fe95d3f8d79829d912
Author: annalu <AnnaLu@dyn-160-39-8-127.dyn.columbia.edu>
Date:   Tue Oct 30 20:20:30 2018 -0400

    Minor changes in binop

commit 78532a437b325695cc39df1421afe308b8f6a519
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Tue Oct 30 19:49:35 2018 -0400

    working on matrix stuff

commit 05fb2ef79cee7e1896da2906aa15db94e757b800
Author: annalu <AnnaLu@dyn-160-39-8-127.dyn.columbia.edu>
Date:   Tue Oct 30 19:47:01 2018 -0400

    Fixed char pretty printing

commit 3da716a60f883b1dee7fcd0786bfe4587c901b13
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Tue Oct 30 18:54:18 2018 -0400

    Fixed local declaration symbol map

    Co-Authored-By: mizutahana <mizutahana@users.noreply.github.com>

commit 7c766c0a427260e66eebb3cb82bc223436959408
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Tue Oct 30 12:33:30 2018 -0400

    updated gitignore for tests/.DS_Store

commit b57bd502afc2834c4af9feed68af18e57d4f6594
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Tue Oct 30 12:21:53 2018 -0400
```

```
    Remove old parser directory that was committed

commit 01017e143ff1539f76bc5df9b980fe51cd8a848d
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Tue Oct 30 09:47:11 2018 -0400

    updated .gitignore and deleted files

commit 22b1a4b8ddec3e8e31077e6cd38f12b2e6003273
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Tue Oct 30 09:39:46 2018 -0400

    ignoring _build and .DS_Store

commit ddd6359de8e4415161499400e4ee02d0127596c4
Author: annalu <AnnaLu@dyn-160-39-9-142.dyn.columbia.edu>
Date:   Tue Oct 30 02:11:41 2018 -0400

    Added pretty printing to AST, binop checks to semant

commit c3546616622c0254f0f6db38587a421ee1402de7
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Tue Oct 30 00:25:30 2018 -0400

    added MatAccess back in returning char for easier testing

commit af299c01bd8db2a0a9f2e15521e4e629bef3f54a
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Tue Oct 30 00:19:16 2018 -0400

    MatAccess commented out, but currently works for accessing elements in a matrix
     called mat. Working on retrieving that matrix given a matrix name

commit fe478299e501566ca6571c7e9b5eb234928f503a
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Mon Oct 29 23:52:48 2018 -0400

    updated matlit so that matrix type has dimensions of matrix just checked

commit f94a2a438dde9aad701f3f98ec4b23770cebb620
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Mon Oct 29 23:47:04 2018 -0400

    commented out specified matrix types, added consolidated matrix

commit 0990ce2ed7200c74b5bb2f6fe34cf24480bd9cde
Merge: 0c31d14 3a9f2e5
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Mon Oct 29 23:31:57 2018 -0400

    Merge branch 'master' of https://github.com/mizutahana/VSCOde into jessica

commit 0c31d1403a60d0198535989dbaf4f59ac0dee403
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Mon Oct 29 23:25:47 2018 -0400

    deleted type checking stuff, matrix return type?

commit 8fee6c67bea8ca4c56f2a0ed1541fb2ecba33fad
Author: Jessica Cheng <jjcheng1998@gmail.com>
```

```
Date:   Mon Oct 29 23:25:23 2018 -0400

    consolidated specified matrix types into one

commit 3a9f2e57a9d7de9966da883e9a8f7773e1aa55c9
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Mon Oct 29 17:12:49 2018 -0400

    Implemented Unop in semant

commit d60f24d2d036f969a9d8bd39bb10b82f13097316
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Mon Oct 29 16:59:30 2018 -0400

    Implemented Tuple in semant

commit 07d48b4ce70d0d15b5bd0ecb022cc98ff37b9c04
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Mon Oct 29 16:55:31 2018 -0400

    Implemented casting in semant

commit ed7015a1ad167314c487d76c78eb378d88e43458
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Mon Oct 29 16:21:20 2018 -0400

    Made sure varnames are declared before they are used in semant

commit 9d173b253d76771e531d2ef9123a3a6b246e4c71
Author: Kenny Yuan <kenny.k.yuan@gmail.com>
Date:   Mon Oct 29 14:16:46 2018 -0400

    Fixed function calls to check number / type of args.

commit 60fffda2bf672ae5294ccc8c6688882becd6b9c1
Author: Kenny Yuan <kenny.k.yuan@gmail.com>
Date:   Mon Oct 29 14:14:12 2018 -0400

    Updated tests to all have main() function. Removed tests with global variables.

commit 9b6e2ecb1c69a663bf7f13be9394a27a39671fe0
Author: Jessica Cheng <jjcheng1998@gmail.com>
Date:   Mon Oct 29 12:46:40 2018 -0400

    first pass on type checking--untested but I have class soon so

commit d6467d2aec0f2e8c41fb60677d96ca80209762c9
Author: Jessica Cheng <Jessica@Jessicas-MacBook-Pro-2.local>
Date:   Mon Oct 29 12:16:29 2018 -0400

    added error, need to add type checking for matrices

commit 7ae3209756f351f65dde0502b8ebd0de79c04118
Author: Jessica Cheng <Jessica@dyn-160-39-141-234.dyn.columbia.edu>
Date:   Mon Oct 29 10:49:43 2018 -0400

    added recursive len_check in matrix that checks to make sure length of each list in
    list of lists is the same

commit ed26a50e8f42b14b13efe96e1652bbcbab53224c
Author: Spencer Yen <spencersyen@gmail.com>
```

```
Date:   Mon Oct 29 00:27:08 2018 -0400

    Implemented if, for, while in stmt

    Co-Authored-By: mizutahana <mizutahana@users.noreply.github.com>

commit 2937a247b703096d858547e2f39fc66b8535405e
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Mon Oct 29 00:08:59 2018 -0400

    Implemented local var declaration (redeclaration allowed)

    Co-Authored-By: mizutahana <mizutahana@users.noreply.github.com>

commit cfadca57951452adb435910862958d1f8fbf6934
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Sun Oct 28 23:37:52 2018 -0400

    Implemented block in semant

    Co-Authored-By: mizutahana <mizutahana@users.noreply.github.com>

commit ac58a153a715fdf071fcab960c79a60ce9b5cb3d
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Sun Oct 28 22:46:36 2018 -0400

    Added exception printing to calc.ml

commit 6203176a8e55e8d5f9381942305cc2e21aa06df6
Merge: 4df49ec ed1969a
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Sun Oct 28 22:25:19 2018 -0400

    Merge branch 'master' of github.com:mizutahana/VSCOde

commit 4df49ec8fd1d02eee1dab19c6dd6c7bd33bf217a
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Sun Oct 28 22:25:18 2018 -0400

    Added tuple as a type

commit ed1969af3fdea03151eee587cf5d50ebccefd140
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Sun Oct 28 21:48:56 2018 -0400

    Added Semant def for local vars

commit 115f18ad9c663d638a2a3844f5492b205f62490e
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Sun Oct 28 17:01:03 2018 -0400

    Added semant defs for Assign, Call, Noexpr, Noassign

commit 07ec5b7974a0a957cb6ceb7b1357e636dd4aa516
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Sun Oct 28 16:37:33 2018 -0400

    Makefile makes with semant, still need to define the semantics for stmts/exprs

commit 90d0b94815a995481f15172be1a08079db9e93b7
Merge: 02b0585 9b5407c
```

```
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Sun Oct 28 15:58:09 2018 -0400

    Merge branch 'master' of github.com:mizutahana/VSCOde

commit 02b05854c540f2d3eac1c7c48df96eb374ce76e6
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Sun Oct 28 15:58:06 2018 -0400

    Changed decl in Parser/AST, Started editing semant (remove global, local)

    In Parser, AST: Changed decl from tuple of fdecl, stmt to list of fdecl (you can't
    have statement outside of function)

    Semant: Not working, but removed globals, local variables and temporarily commented
    out built_in_decls. To do:
    Co-Authored-By: mizutahana <mizutahana@users.noreply.github.com>

commit 9b5407c0654999917949bcaee718b69fad4c53b5
Author: Jessica Cheng <Jessica@jessicas-mbp-2.lan>
Date:   Sun Oct 28 14:26:55 2018 -0400

    updated gitignore to match new directory structure

commit cf9c901f5b2fa79b38b19655779f8f1c5b1aa478
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Sun Oct 28 14:23:45 2018 -0400

    Reorganized directory, removed microc

commit a1b82ea4bd37bf36b4e2f5a50ae550a60d779bc9
Author: Kenneth Yuan <kenny.k.yuan@gmail.com>
Date:   Sat Oct 27 14:06:38 2018 -0400

    Wrote series of test files to check our implementation of semant.

commit 93a4bbabecb40efd2930efdf6c815372318db5e6
Author: Kenny Yuan <kenny.k.yuan@gmail.com>
Date:   Mon Oct 22 13:48:33 2018 -0400

    Updated Makefile and top-level to be compatible with semant checking. In order for
    this to work, we need to replace ast.mli with ast.ml.

commit a08f907cd1053aa2275cccbc770eaeac14634156
Merge: 1a7e113 a0e419d
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Mon Oct 22 10:48:07 2018 -0400

    Merge branch 'master' of github.com:mizutahana/VSCOde

commit 1a7e113fe713c4fa3a1d8c9364c93cd9a164bc2a
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Mon Oct 22 10:48:04 2018 -0400

    Added string concat with caret

commit a0e419d81f54f50d9ca084d53486927d3530eef6
Author: annalu <AnnaLu@dyn-160-39-8-210.dyn.columbia.edu>
Date:   Mon Oct 22 02:41:54 2018 -0400

    Removed extranneous tokens from ast.mll
```

```
commit 51076ea47e6f2326d87835144f43bf2cd29225ec
Author: annalu <AnnaLu@Annas-MacBook-Air-3.local>
Date:   Mon Oct 15 23:00:30 2018 -0400

    Added menhir as Makefile rule

commit 2875c930238617bc6d822f4374743b4d7513175e
Author: annalu <AnnaLu@Annas-MacBook-Air-3.local>
Date:   Mon Oct 15 22:05:30 2018 -0400

    Removed extranneous tokens

commit 1f4ae911ab894230de706a7c3fa8ca31983dccdb
Merge: cc34585 eebec75
Author: Jessica Cheng <Jessica@Jessicas-MacBook-Pro-2.local>
Date:   Mon Oct 15 21:56:47 2018 -0400

    Merge branch 'master' of https://github.com/mizutahana/VSCOde

commit eebec7504846200089f3040ed194356705469dee
Author: annalu <AnnaLu@Annas-MacBook-Air-3.local>
Date:   Mon Oct 15 21:54:30 2018 -0400

    Fixed parentheses precedence, which was not functional previously

commit 610e4ae857e9191d53f4fcd720785b130ac09a29
Author: annalu <AnnaLu@Annas-MacBook-Air-3.local>
Date:   Mon Oct 15 21:42:35 2018 -0400

    Added parentheses for precedence

commit dad27f62ff356084ec1a956ee3dc782924060b2c
Merge: 98f5d9b ed960a0
Author: annalu <AnnaLu@Annas-MacBook-Air-3.local>
Date:   Mon Oct 15 21:32:18 2018 -0400

    Merge branch 'master' of https://github.com/mizutahana/VSCOde

commit 98f5d9b9c93d9bcce2760b86e1496acf6cd65442
Author: annalu <AnnaLu@Annas-MacBook-Air-3.local>
Date:   Mon Oct 15 21:32:12 2018 -0400

    Added parentheses for precedence

commit cc345859450bcfb57e83f46a1f1ecb1d515ffa97
Merge: 3baee64 ed960a0
Author: Jessica Cheng <Jessica@Jessicas-MacBook-Pro-2.local>
Date:   Mon Oct 15 21:30:48 2018 -0400

    Merge branch 'master' of https://github.com/mizutahana/VSCOde

commit ed960a0b968567538c5bfaa4997a47df9997bac9
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Mon Oct 15 21:27:52 2018 -0400

    Removed tuple type

commit 3baee64fa915db6ec8f6441ce59c25bf45e7521e
Merge: ce27177 affff18
Author: Jessica Cheng <Jessica@Jessicas-MacBook-Pro-2.local>
```

```
Date:   Mon Oct 15 21:26:05 2018 -0400

    Merge branch 'master' of https://github.com/mizutahana/VSCOde

commit affff18c5f14590488ed9d74f04ed950d2582606
Author: annalu <AnnaLu@Annas-MacBook-Air-3.local>
Date:   Mon Oct 15 21:20:12 2018 -0400

    Fixed merge conflict

commit bd58fc8ea9f523f182f254af0b720ca6aebe101c
Merge: 5767f94 760efc1
Author: annalu <AnnaLu@Annas-MacBook-Air-3.local>
Date:   Mon Oct 15 21:18:15 2018 -0400

    Added function calls on objects

commit 5767f94adf603c859d3e533a57b531df69a34be4
Author: annalu <AnnaLu@Annas-MacBook-Air-3.local>
Date:   Mon Oct 15 21:15:54 2018 -0400

    Added function calls on objects

commit ce27177ab47ffffb6b44cb4b2ffd507fef5415ac
Merge: 3361acd 760efc1
Author: Jessica Cheng <Jessica@Jessicas-MacBook-Pro-2.local>
Date:   Mon Oct 15 21:09:13 2018 -0400

    untracking parser.output

commit 3361acdaf972076d04b658fd28142faa4645ff3f
Author: Jessica Cheng <Jessica@Jessicas-MacBook-Pro-2.local>
Date:   Mon Oct 15 21:03:38 2018 -0400

    remove parser.output from being tracked

commit 760efc1c7e6a315ba4533b12169a421236c89769
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Mon Oct 15 21:00:22 2018 -0400

    Removed multiassign

commit 88cc03497c3f1a61e1242606b7d8ba612adb1126
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Mon Oct 15 20:49:34 2018 -0400

    Better designed cast implementation

commit e84c359cecfdf72cf56a6ca97f7e8fa1c06305e9
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Mon Oct 15 20:12:15 2018 -0400

    Added casting

commit 4d8b7f12f284b23579134f984ec740a94fa59b2a
Merge: 7ba5f40 3050079
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Mon Oct 15 17:27:44 2018 -0400

    Merge branch 'master' of github.com:mizutahana/VSCOde
```

```
commit 7ba5f40b31f2300e22199b58d1b385b2661b7ab5
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Mon Oct 15 17:27:21 2018 -0400

    Added tuple type and tuple literal, statement multiassign, commented out expr
    multiassign

commit 305007908259b58050860e382e9c0b4825b770fa
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Mon Oct 15 17:21:51 2018 -0400

    Removed parser.output

commit 79e81347fda50900856c939a4dfb94693226f149
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Mon Oct 15 11:51:27 2018 -0400

    Multilocal assign is valid, commented out multiassign for expr and cast

commit 50c6d18a6851541bd0d944a8de88db96dc122400
Merge: 8e138f0 8759fd1
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Mon Oct 15 11:29:03 2018 -0400

    Merge branch 'master' of github.com:mizutahana/VSCOde

commit 8e138f07fb038e1983e82ab6f737323f97e74365
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Mon Oct 15 11:28:28 2018 -0400

    Remove MultiAssign for now

commit 8759fd1c366f9061841ed85d958b1dc00e5ce4cb
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Mon Oct 15 11:27:38 2018 -0400

    Fixed code for char literal

commit e32ffade4e4e3ab215d2bc6e88eb0225b32972ae
Author: Jessica Cheng <Jessica@dyn-160-39-140-236.dyn.columbia.edu>
Date:   Mon Oct 15 11:17:50 2018 -0400

    ignoring parser output

commit 154bb70a2625b4953f155bcd4e594dcd433495df
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Mon Oct 15 11:13:54 2018 -0400

    Multiassign (has SR conflicts)

commit 8643e13a1703ad3aae16539fde40795b1a64275f
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Mon Oct 15 10:55:40 2018 -0400

    Added casting

commit 81d19f3a1c98f2a42487660a6fc645bf27225981
Author: Jessica Cheng <Jessica@dyn-160-39-140-236.dyn.columbia.edu>
Date:   Mon Oct 15 10:37:34 2018 -0400

    changed + to * in double (can support .1 instead of needing 0.1)
```

```
commit 8c73367d5f47fd61d978e893272699494a607ba5
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Mon Oct 15 10:26:40 2018 -0400

    Removed unnecessary comments

commit 51255495591a6216b8a42099019b55bb1574c7f7
Merge: ee03c83 6e47de0
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Mon Oct 15 10:24:29 2018 -0400

    Resolved merge conflict

commit ee03c83c55c9127e949415f092e0e22a00ff2212
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Mon Oct 15 10:21:34 2018 -0400

    Removed comments

commit 6e47de06233cb74ec4c0c9d011fc0ef380799212
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Mon Oct 15 10:20:52 2018 -0400

    Added single comment

commit 6284c13c822dadb3986fcce977a5f4f46b6ce86e
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Sun Oct 14 11:16:19 2018 -0400

    Remove todo note for error on function call

commit 8753d57074c0aed25c91cbaf182c8c234c1ab3ff
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Sun Oct 14 11:04:50 2018 -0400

    Implemented matrix and tuple in Parser, implemented ast, cleaned up Scanner

    To do: function call with no param currently fails (e.g. test()). Look at line
    119/actuals_opt in parser.mly

commit 098fe3dfc6bd0ce7d11bae6d93f70ff3d23df7fb
Author: annalu <AnnaLu@dyn-160-39-9-150.dyn.columbia.edu>
Date:   Thu Oct 11 20:07:20 2018 -0400

    Updated matrix access syntax

commit 1269548c1883eb9773a0d4de9c2309225b5732ee
Author: annalu <AnnaLu@dyn-160-39-9-150.dyn.columbia.edu>
Date:   Thu Oct 11 19:59:14 2018 -0400

    Fixed some ast.mli rules, but still has errors

commit 12cf34eeeaa93da486c007b44b441fdde566ac51
Author: annalu <AnnaLu@dyn-160-39-9-150.dyn.columbia.edu>
Date:   Thu Oct 11 19:34:38 2018 -0400

    Added matrix syntax to parser. Fixed comma shift-reduce errors

commit f706646bbbca51310fd28848827172265 89f1581
Author: Hana Mizuta <mizuta.hana@columbia.edu>
```

```
Date:   Thu Oct 11 09:29:47 2018 -0400

    Removed duplicate in parser, removed 31R/R and 2 never reduced errors

commit 8b0beb4d5062377db13140472a1544a4278bedb6
Author: Kenneth Yuan <kenny.k.yuan@gmail.com>
Date:   Thu Oct 11 09:20:52 2018 -0400

    Fixed 34 shift/reduce conflicts by adding associativity rule for COMMA.

commit 35aa5871c07f73cb6ba7aa6000bad51a8b705811
Author: Jessica Cheng <Jessica@dyn-160-39-140-204.dyn.columbia.edu>
Date:   Thu Oct 11 00:11:11 2018 -0400

    parser, scanner, ast without matrix implemented and with parser.output shift/reduce
     and reduce/reduce errors

commit 5cb87a0cc0443a9ebf4fb3998734b421e18baf28
Author: Jessica Cheng <Jessica@dyn-160-39-140-204.dyn.columbia.edu>
Date:   Thu Oct 11 00:10:04 2018 -0400

    35 shift/reduce conflicts. 29 reduce/reduce conflicts.

commit aee456f7fa2dd889132a2585b539cd16445dda64
Author: Spencer Yen <spencersyen@gmail.com>
Date:   Wed Oct 10 23:32:13 2018 -0400

    Added mod to parser

commit f9b05b0606b8d18069cb3c13fa349279f4037df0
Merge: 1ffd4aa b3a6966
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Wed Oct 10 23:28:27 2018 -0400

    Merge branch 'master' of https://github.com/mizutahana/PLTEA

commit 1ffd4aa6f3f06ddd1619f76171923aafe1995307
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Wed Oct 10 23:28:14 2018 -0400

    Parser in progress...

commit b3a6966c5e823c06c207160cbbba8118175057e8
Merge: a5aed9d dcea5b4
Author: Jessica Cheng <Jessica@dyn-160-39-140-204.dyn.columbia.edu>
Date:   Wed Oct 10 22:00:10 2018 -0400

    Merge branch 'master' of https://github.com/mizutahana/PLTEA

commit a5aed9d0b086f655c95a904a4391fc19a0070c3b
Author: Jessica Cheng <Jessica@dyn-160-39-140-204.dyn.columbia.edu>
Date:   Wed Oct 10 21:59:38 2018 -0400

    gitignore file, currently has files to ignore after compiling parser

commit dcea5b44a2dfdb31564ab6ffc816ebd8be61368e
Author: Kenneth Yuan <kenny.k.yuan@gmail.com>
Date:   Wed Oct 10 21:57:16 2018 -0400

    Implemented simple tester to check if input can get parsed or not.
```

```
commit 610935b62a9d3ca56a63390bf563d92d5d64c62f
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Wed Oct 10 21:51:18 2018 -0400

    Restructured directory, creating parser folder

commit b5df67fdf2caf5b4e26ff2579605c0d1425b6cf9
Author: Hana Mizuta <mizuta.hana@columbia.edu>
Date:   Wed Oct 10 21:33:38 2018 -0400

    Added symbols to scanner.mll

commit b0142b8c46e4f6d6c7f97821f9f94cbc0ffe9e98
Author: mizutahana <hm2694@columbia.edu>
Date:   Mon Sep 10 23:41:30 2018 -0400

    Initial commit
```

9.2 testall.sh

```
# Path to the LLVM interpreter
LLI="lli"
#LLI="/usr/local/opt/llvm/bin/lli"

# Path to the LLVM compiler
LLC="llc"

# Path to the C compiler
CC="cc"

# Path to the microc compiler.  Usually "./microc.native"
# Try "_build/microc.native" if ocamlbuild was unable to create a symbolic link.
VSCODE="./vscode.native"
#MICROC="_build/microc.native"

# Set time limit for all operations
ulimit -t 500

globallog=testall.log
rm -f $globallog
error=0
globalerror=0

keep=0

Usage() {
    echo "Usage: testall.sh [options] [.vsc files]"
    echo "-k    Keep intermediate files"
    echo "-h    Print this help"
    exit 1
}

SignalError() {
    if [ $error -eq 0 ] ; then
    echo "FAILED [U+FFFD][U+FFFD]"
    error=1
    fi
    echo "  $1"
}
```

```
# Compare <outfile> <reffile> <difffile>
# Compares the outfile with reffile.  Differences, if any, written to difffile
Compare() {
    generatedfiles="$generatedfiles $3"
    echo diff -b $1 $2 ">" $3 1>&2
    diff -b "$1" "$2" > "$3" 2>&1 || {
    SignalError "$1 differs"
    echo "FAILED $1 differs from $2" 1>&2
    }
}

# Run <args>
# Report the command, run it, and report any errors
Run() {
    echo $* 1>&2
    eval $* || {
         SignalError "$1 failed on $*"
         return 1
    }
}

# RunFail <args>
# Report the command, run it, and expect an error
RunFail() {
    echo $* 1>&2
    eval $* && {
    SignalError "failed: $* did not report an error"
    return 1
    }
    return 0
}

Check() {
    error=0
    basename=‘echo $1 | sed ’s/.*\\///
                             s/.vsc//’‘
    reffile=‘echo $1 | sed ’s/.vsc$//’‘
    basedir="‘echo $1 | sed ’s/\/[^\/]*$//’‘/."

    echo -n "$basename..."

    echo 1>&2
    echo "###### Testing $basename" 1>&2

    generatedfiles=""

    # generatedfiles="$generatedfiles ${basename}.ll ${basename}.s ${basename}.exe
    ${basename}.out" &&
    generatedfiles="$generatedfiles ${basename}.ir ${basename}.ir.s ${basename}.exe
    ${basename}.out" &&
    Run "./link.sh" "${reffile}" &&
    # Run "$VSCODE" "$1" ">" "${basename}.ll" &&
    # Run "$LLC" "-relocation-model=pic" "${basename}.ll" ">" "${basename}.s" &&
    # Run "$CC" "-o" "${basename}.exe" "${basename}.s" &&
    Run "./${basename}.exe" ">" "${basename}.out" &&
    Compare ${basename}.out ${reffile}.out ${basename}.diff

    # Report the status and clean up the generated files

    if [ $error -eq 0 ] ; then
```

```
        if [ $keep -eq 0 ] ; then
            rm -f $generatedfiles
        fi
        echo "OK [U+FFFD][U+FFFD] [U+FFFD][U+FFFD]"
        echo "###### SUCCESS" 1>&2
        else
        echo "###### FAILED" 1>&2
    globalerror=$error
    fi
}

CheckFail() {
    error=0
    basename=‘echo $1 | sed ’s/.*\\///
                            s/.vsc//’‘
    reffile=‘echo $1 | sed ’s/.vsc$//’‘
    basedir="‘echo $1 | sed ’s/\/[^\/]*$//’‘/."

    echo -n "$basename..."

    echo 1>&2
    echo "###### Testing $basename" 1>&2

    generatedfiles=""

    generatedfiles="$generatedfiles ${basename}.err ${basename}.diff" &&
    Run "$VSCODE" "<" $1 ">" "${basename}.err" &&
    Compare ${basename}.err ${reffile}.err ${basename}.diff

    # Report the status and clean up the generated files

    if [ $error -eq 0 ] ; then
        if [ $keep -eq 0 ] ; then
            rm -f $generatedfiles
        fi
        echo "OK [U+FFFD][U+FFFD] [U+FFFD][U+FFFD]"
        echo "###### SUCCESS" 1>&2
        else
        echo "###### FAILED" 1>&2
    globalerror=$error
    fi
}

while getopts kdpsh c; do
    case $c in
    k) # Keep intermediate files
        keep=1
        ;;
    h) # Help
        Usage
        ;;
    esac
done

shift ‘expr $OPTIND - 1‘

LLIFail() {
  echo "Could not find the LLVM interpreter \"$LLI\"."
  echo "Check your LLVM installation and/or modify the LLI variable in testall.sh"
  exit 1
}
```

```
which "$LLI" >> $globallog || LLIFail

#if [ ! -f printbig.o ]
#then
#    echo "Could not find printbig.o"
#    echo "Try \"make printbig.o\""
#    exit 1
#fi

if [ $# -ge 1 ]
then
    files=$@
else
    files="tests/test-*.vsc tests/fail-*.vsc"
fi

for file in $files
do
    case $file in
    *test-*)
        Check $file 2>> $globallog
        ;;
    *fail-*)
        CheckFail $file 2>> $globallog
        ;;
    *)
        echo "unknown file type $file"
        globalerror=1
        ;;
    esac
done

exit $globalerror
```

9.3 Full Test Suite

fail-add1.vsc

```
func main() -> int {
    3 + "hi";
    return 0;
}
```

fail-add1.err

```
Error with semant:
Failure("illegal binary operator")
```

fail-add2.vsc

```
func main() -> int {
    3 + 3.0;
    return 0;
}
```

fail-add2.err

```
Error with semant:
Failure("illegal binary operator")
```

fail-assign1.vsc

```
func main() -> void
{
  double y = 3;
  return;
}
```

fail-assign1.err

```
Error with semant:
Failure("Local var type does not match")
```

fail-assign2.vsc

```
func main() -> void
{
  int k = 3 + "hello";
  return;
}
```

fail-assign2.err

```
Error with semant:
Failure("illegal binary operator")
```

fail-assign3.vsc

```
func main() -> void
{
  int y = "hi";
  return;
}
```

fail-assign3.err

```
Error with semant:
Failure("Local var type does not match")
```

fail-assign4.vsc

```
func main() -> void
{
  int y = 4;
  int z = y[0,0];
  return;
}
```

fail-assign4.err

```
Error with semant:
Failure("Illegal matrix access")
```

fail-assign5.vsc

```
func main() -> void
{
  int y = 4;
  bool z = true;
  int x = y + z;
  return;
}
```

fail-assign5.err

```
Error with semant:
Failure("illegal binary operator")
```

fail-assign6.vsc

```
func main() -> void
{
  int k = return(3);
}
```

fail-assign6.err

```
Error with parsing
```

fail-assign7.vsc

```
func main() -> void
{
  int for = 3;
  return;
}
```

fail-assign7.err

```
Error with parsing
```

fail-assign8.vsc

```
func main() -> void
{
  int i;
  bool b;

  i = 42;
  i = 10;
  b = true;
  b = false;
  i = false;

  return;
}
```

fail-assign8.err

```
Error with semant:
Failure("Check_assign failed with illegal assignment")
```

fail-call1.vsc

```
func main() -> void
{
  int k = 3;
  k(10);
  return;
}
```

fail-call1.err

```
Error with semant:
Failure("unrecognized function k")
```

fail-call2.vsc

```
func main() -> int
{
  int i = add(1, 2);
  return i;
}
```

fail-call2.err

```
Error with semant:
Failure("unrecognized function add")
```

fail-casting.vsc

```
func main() -> void
{
  char c = 'a';
  int i = (int) c;
  return;
}
```

fail-casting.err

```
Error with parsing
```

fail-for1.vsc

```
func main() -> int
{
  int i;
  for ( ; true ; ) {} /* OK: Forever */

  for (i = 0 ; i < 10 ; i = i + 1) {
    if (i == 3) return 42;
```

```
  }

  for (j = 0; i < 10 ; i = i + 1) {} /* j undefined */

  return 0;
}
```

fail-for1.err

```
Error with semant:
Failure("undeclared identifier j")
```

fail-for2.vsc

```
func main() -> int
{
  int i;

  for (i = 0; j < 10 ; i = i + 1) {} /* j undefined */

  return 0;
}
```

fail-for2.err

```
Error with semant:
Failure("Variable not found: j")
```

fail-for3.vsc

```
func main() -> int
{
  int i;

  for (i = 0; i ; i = i + 1) {} /* i is an integer, not Boolean */

  return 0;
}
```

fail-for3.err

```
Error with semant:
Failure("expected Boolean expression")
```

fail-for4.vsc

```
func main() -> int
{
  int i;

  for (i = 0; i < 10 ; i = j + 1) {} /* j undefined */

  return 0;
}
```

fail-for4.err

```
Error with semant:
Failure("Variable not found: j")
```

fail-for5.vsc

```
func main() -> int
{
  int i;

  for (i = 0; i < 10 ; i = i + 1) {
    foo(); /* Error: no function foo */
  }

  return 0;
}
```

fail-for5.err

```
Error with semant:
Failure("unrecognized function foo")
```

fail-for6.vsc

```
func main() -> void
{
  for (int i = 0; i < 5; i = i + 1) {
    print("FAIL");
  }
  return;
}
```

fail-for6.err

```
Error with parser
```

fail-func1.vsc

```
func foo() -> int { return 0; }

func bar() -> int { return 1; }

func bar() -> void { return; } /* Error: duplicate function bar */

func main() -> void { return; }
```

fail-func1.err

```
Error with semant:
Failure("Adding func failed with duplicate function bar")
```

fail-func2.vsc

```
func foo(int a, bool b, int c) -> void { return; }
```

```
func bar(int a, bool b, int a) -> void { return; } /* Error: duplicate formal a in bar */

func main() -> void { return; }
```

fail-func2.err

```
Error with semant:
Failure("duplicate formal a")
```

fail-func3.vsc

```
func foo(int a, bool b, int c) -> int { return 1; }

func bar(int a, void b, int c) -> void { return; } /* Error: illegal void formal b */

func main() -> void { return; }
```

fail-func3.err

```
Error with semant:
Failure("illegal void formal b")
```

fail-func4.vsc

```
func foo() -> int { return 1; }

func bar() -> void { return; }

fun print() -> int { return 2; } /* Should not be able to define print */

func main() -> void { return; }
```

fail-func4.err

```
Error with parsing
```

fail-func5.vsc

```
func bar() -> void
{
  int a;
  void b; /* Error: illegal void local b */
  bool c;

  return;
}
```

fail-func5.err

```
Error with semant:
Failure("unrecognized function main")
```

fail-func6.vsc

```
func foo(int a, bool b) -> void
```

```
{
  return;
}

func main() -> void
{
  foo(42, true);
  foo(42); /* Wrong number of arguments */
  return;
}
```

fail-func6.err

```
Error with semant:
Failure("expecting 2 arguments in foo")
```

fail-func7.vsc

```
func foo(int a, bool b) -> void
{
  return;
}

func main() -> void
{
  foo(42, true);
  foo(42, true, false); /* Wrong number of arguments */
  return;
}
```

fail-func7.err

```
Error with semant:
Failure("expecting 2 arguments in foo")
```

fail-func8.vsc

```
func foo(int a, bool b) -> void
{
  return;
}

func bar() -> void
{
  return;
}

func main() -> void
{
  foo(42, true);
  foo(42, bar()); /* int and void, not int and bool */
  return;
}
```

fail-func8.err

```
Error with semant:
```

```
Failure("Check_assign failed with illegal actual argument")
```

fail-func9.vsc

```
func foo(int a, bool b) -> void
{
  return;
}

func main() -> void
{
  foo(42, true);
  foo(42, 42); /* Fail: int, not bool */
  return;
}
```

fail-func9.err

```
Error with semant:
Failure("Check_assign failed with illegal actual argument")
```

fail-if1.vsc

```
func main() -> int
{
  if ("hello") {
    return 5;
  }
  return 1;
}
```

fail-if1.err

```
Error with semant:
Failure("expected Boolean expression")
```

fail-if2.vsc

```
func main() -> void
{
  if (true) {
    foo; /* Error: undeclared variable */
  }
  return;
}
```

fail-if2.err

```
Error with semant:
Failure("Variable not found: foo")
```

fail-if3.vsc

```
func main() -> void
{
```

```
  if (true) {
    42;
  } else {
    bar; /* Error: undeclared variable */
  }
  return;
}
```

fail-if3.err

```
Error with semant:
Failure("Variable not found: bar")
```

fail-if4.vsc

```
func main() -> void
{
  if (true) {}
  if (false) {} else {}
  if (42) {} /* Error: non-bool predicate */
  return;
}
```

fail-if4.err

```
Error with semant:
Failure("expected Boolean expression")
```

fail-matrix1.vsc

```
func main() -> int
{
  matrix m = [ 1.0,1.0,1.0 ] + [ 1.0,1.0 ];
  return 0;
}
```

fail-matrix1.err

```
Error with semant:
Failure("Cannot add/subtract matrices with different dimensions")
```

fail-matrix2.vsc

```
func main() -> matrix
{
  matrix m = [ 1.0,2.0,"hi"; ];
  return m;
}
```

fail-matrix2.err

```
Error with parsing
```

fail-matrix3.vsc

```
func main() -> matrix
{
  matrix m = [ 1,2.0,3; ];
  return m;
}
```

fail-matrix3.err

```
Error with parsing
```

fail-matrix4.vsc

```
func main() -> int
{
  matrix m = [ 1.0, 1.0 ];
  return m[9,9];
}
```

fail-matrix4.err

```
Error with semant:
Failure("wrong return type")
```

fail-matrix5.vsc

```
func main() -> int {
  matrix m = [ 1.0,1.0,1.0 ];
  return m[0,0];
}
```

fail-matrix5.err

```
Error with semant:
Failure("wrong return type")
```

fail-return1.vsc

```
func main() -> int
{
  return true; /* Should return int */
}
```

fail-return1.err

```
Error with semant:
Failure("wrong return type")
```

fail-return2.vsc

```
func foo() -> void
{
  if (true) return 42; /* Should return void */
  else return;
}
```

```
func main() -> int
{
  return 42;
}
```

fail-return2.err

```
Error with semant:
Failure("wrong return type")
```

fail-while1.vsc

```
func main() -> void
{
  int i;

  while (true) {
    i = i + 1;
  }

  while (42) { /* Should be boolean */
    i = i + 1;
  }

  return;
}
```

fail-while1.err

```
Error with semant:
Failure("expected Boolean expression")
```

fail-while2.vsc

```
func main() -> void
{
  int i;

  while (true) {
    i = i + 1;
  }

  while (true) {
    foo(); /* foo undefined */
  }

  return;
}
```

fail-while2.err

```
Error with semant:
Failure("unrecognized function foo")
```

test-add1.vsc

```
func add(int x, int y) -> int {
  return x + y;
}

func main() -> int {
  print( add(17, 25) );
  return 0;
}
```

test-add1.out

```
42
```

test-arith1.vsc

```
func main() -> int {
  print(39 + 3);
  return 0;
}
```

test-arith1.out

```
42
```

test-arith2.vsc

```
func main() -> int {
  print(1.82 + 342.00);
  return 0;
}
```

test-arith2.out

```
343.820
```

test-arith3.vsc

```
func foo(int a) -> int {
  return a;
}

func main() -> int {
  int a;
  a = 42;
  a = a + 5;
  print(a);
  return 0;
}
```

test-arith3.out

```
47
```

test-assign1.vsc

```
func main() -> int {
    int x;
    x = 5;
    if (x == 5) {print("assign after noassign works!"); }
    return 0;
}
```

test-assign1.out

```
assign after noassign works!
```

test-binop-add-matrix-1.vsc

```
func main() -> int {
    matrix m = [1.0, 2.0; 3.0, 4.0];
    matrix n = [2.0, 3.0; 5.0, 6.0];
    matrix o = m + n;
    print(o[1,1]);
    return 0;
}
```

test-binop-add-matrix-1.out

```
10.000
```

test-binop-mul-matrix-1.vsc

```
func main() -> int {
  matrix a = [2.0,5.0;3.0,4.0];
  matrix b = [3.0;4.0];
  matrix c = a * b;
  print(c[0,0]);
}
```

test-binop-mul-matrix-1.out

```
26.000
```

test-binop-scalarmul-matrix-1.vsc

```
func main() -> int {
    matrix m = [12.0, 4.0, 7.0; 1.0, 9.2, 9.3];
    matrix n = m * 3.5;
    double x = 2.5;
    matrix o = m * x;
    matrix a = 3.5 * m;
    matrix b = x * m;
    print(n[0,1]);
    print(o[0,1]);
    print(a[0,1]);
    print(b[0,1]);
    return 0;
}
```

test-binop-scalarmul-matrix-1.out

```
14.000
10.000
14.000
10.000
```

test-binop1.vsc

```
func main() -> int {
  int x = 17 + 25;
  if (x == 42) {
    print("add works!");
  } else {
    print("add broken :(");
  }
  return 0;
}
```

test-binop1.out

```
add works!
```

test-binop2.vsc

```
func main() -> int {
  double a = 2.0;
  double b = 3.0;
  double c = a / b;
  double d = 4.0 / 6.0;
  if (c == d) {
    print("double divide works!");
  }
  return 0;
}
```

test-binop2.out

```
double divide works!
```

test-binop3.vsc

```
func main() -> int {
  bool a = true;
  bool b = false;
  bool c = true;
  if ((a != b) && (a == c)) {
    print("binop works!");
  }
  return 0;
}
```

test-binop3.out

```
binop works!
```

test-casting2.vsc

```
func main() -> int {
  double d = 10.7;
  int i = (int) d;
  print(i);
  return 0;
}
```

test-casting2.out

```
10
```

test-casting4.vsc

```
func main() -> int {
  bool b = true;
  print(b);
  return 0;
}
```

test-casting4.out

```
1
```

test-fib.vsc

```
func fib(int x) -> int {
  if (x < 2) return 1;
  return fib(x-1) + fib(x-2);
}

func main() -> int {
  print(fib(0));
  print(fib(1));
  print(fib(2));
  print(fib(3));
  print(fib(4));
  print(fib(5));
  return 0;
}
```

test-fib.out

```
1
1
2
3
5
8
```

test-for1.vsc

```
func main() -> int {
  int i;
  for (i = 0 ; i < 5 ; i = i + 1) {
    print(i);
  }
```

```
  print(42);
  return 0;
}
```

test-for1.out

```
0
1
2
3
4
42
```

test-for2.vsc

```
func main() -> int {
  int i;
  i = 0;
  for ( ; i < 5; ) {
    print(i);
    i = i + 1;
  }
  print(42);
  return 0;
}
```

test-for2.out

```
0
1
2
3
4
42
```

test-func1.vsc

```
func add(int a, int b) -> int {
  return a + b;
}

func main() -> int {
  int a;
  a = add(39, 3);
  print(a);
  return 0;
}
```

test-func1.out

```
42
```

test-func2.vsc

```
func fun(int x, int y) -> int {
  return 0;
```

```
}

func main() -> int {
  int i;
  i = 1;

  fun(i = 2, i = i + 1);

  print(i);
  return 0;
}
```

test-func2.out

```
2
```

test-func3.vsc

```
func printem(int a, int b, int c, int d) -> void {
  print(a);
  print(b);
  print(c);
  print(d);
}

func main() -> int {
  printem(42,17,192,8);
  return 0;
}
```

test-func3.out

```
42
17
192
8
```

test-func4.vsc

```
func add(int a, int b) -> int {
  int c;
  c = a + b;
  return c;
}

func main() -> int {
  int d;
  d = add(52, 10);
  print(d);
  return 0;
}
```

test-func4.out

```
62
```

test-func5.vsc

```
func foo(int a) -> int {
  return a;
}

func main() -> int {
  return 0;
}
```

test-func5.out

test-func6.vsc

```
func foo() -> void {}

func bar(int a, bool b, int c) -> int { return a + c; }

func main() -> int {
  print(bar(17, false, 25));
  return 0;
}
```

test-func6.out

```
42
```

test-func7.vsc

```
func foo(int c) -> int {
  return (c + 42);
}

func main() -> int {
  int a = foo(73);
  print(a);
  return 0;
}
```

test-func7.out

```
115
```

test-func8.vsc

```
func foo(int a) -> int {
  int temp = a + 3;
  print(temp);
  return 0;
}

func main() -> int {
  foo(40);
  return 0;
}
```

test-func8.out

```
43
```

test-gcd.vsc

```
func gcd(int a, int b) -> int {
  while (a != b) {
    if (a > b) a = a - b;
    else b = b - a;
  }
  return a;
}

func main() -> int {
  print(gcd(2,14));
  print(gcd(3,15));
  print(gcd(99,121));
  return 0;
}
```

test-gcd.out

```
2
3
11
```

test-gcd2.vsc

```
func gcd(int a, int b) -> int {
  while (a != b)
    if (a > b) a = a - b;
    else b = b - a;
  return a;
}

func main() -> int {
  print(gcd(14,21));
  print(gcd(8,36));
  print(gcd(99,121));
  return 0;
}
```

test-gcd2.out

```
7
4
11
```

test-hello.vsc

```
func main() -> int {
  print("Hello World! My name is VSCOde. I am sentient.");
  return 0;
}
```

test-hello.out

```
Hello World! My name is VSCOde. I am sentient.
```

test-if1.vsc

```
func main() -> int {
  if (true) print(42);
  print(17);
  return 0;
}
```

test-if1.out

```
42
17
```

test-if2.vsc

```
func main() -> int {
  if (true) print(42); else print(8);
  print(17);
  return 0;
}
```

test-if2.out

```
42
17
```

test-if3.vsc

```
func main() -> int {
  if (false) print((string) 42);
  print(17);
  return 0;
}
```

test-if3.out

```
17
```

test-if4.vsc

```
func main() -> int {
  if (false) print(42); else print(8);
  print(17);
  return 0;
}
```

test-if4.out

```
8
```

```
17
```

test-if5.vsc

```
func cond(bool b) -> int {
  int x;
  if (!b)
    x = 42;
  else
    x = 17;
  return x;
}

func main() -> int {
 print(cond(false));
 print(cond(true));
 return 0;
}
```

test-if5.out

```
42
17
```

test-image-blur.vsc

```
func main() -> int {
    image im = blur("red.jpg");
    matrix r = im.red;
    matrix g = im.green;
    matrix b = im.blue;
    print(r[0,0]);
    print(g[0,0]);
    print(b[0,0]);
    return 0;
}
```

test-image-blur.out

```
236.000
34.000
50.000
```

test-image-brighten.vsc

```
func main() -> int {
    image im = brighten("red.jpg");
    matrix r = im.red;
    matrix g = im.green;
    matrix b = im.blue;
    print(r[0,0]);
    print(g[0,0]);
    print(b[0,0]);
    return 0;
}
```

test-image-brighten.out

```
236.000
34.000
50.000
```

test-image-dim.vsc

```
func main() -> int {
    matrix size = dim("red.jpg");
    print(size[0,0]);
    print(size[0,1]);
    return 0;
}
```

test-image-dim.out

```
629.000
700.000
```

test-image-edgedetect.vsc

```
func main() -> int {
    image im = edgedetect("red.jpg");
    matrix r = im.red;
    matrix g = im.green;
    matrix b = im.blue;
    print(r[0,0]);
    print(g[0,0]);
    print(b[0,0]);
    return 0;
}
```

test-image-edgedetect.out

```
0.000
0.000
0.000
```

test-image-grayscale.vsc

```
func main() -> int {
    image im = grayscale("red.jpg");
    matrix r = im.red;
    matrix g = im.green;
    matrix b = im.blue;
    print(r[0,0]);
    print(g[0,0]);
    print(b[0,0]);
    return 0;
}
```

test-image-grayscale.out

```
62.000
62.000
62.000
```

test-image-special.vsc

```
func main() -> int {
    image im = stephenedwards("red.jpg");
    matrix r = im.red;
    matrix g = im.green;
    matrix b = im.blue;
    print(r[0,0]);
    print(g[0,0]);
    print(b[0,0]);
    return 0;
}
```

test-image-special.out

```
73.000
61.000
47.000
```

test-image-load1.vsc

```
func main() -> int {
    image im = load("red.jpg");
    return 0;
}
```

test-image-load1.out

test-image-load2.vsc

```
func main() -> int {
    image im = load("red.jpg");
    matrix r = im.red;
    matrix g = im.green;
    matrix b = im.blue;
    print(r[0,0]);
    print(g[0,0]);
    print(b[0,0]);
    return 0;
}
```

test-image-load2.out

```
236.000
34.000
50.000
```

test-image-local1.vsc

```
func foo(bool b) -> int {
  int i;

  i = 42;
  print((i + i));
```

```
  return 0;
}

func main() -> int {
  foo(true);
  return 0;
}
```

test-image-local1.out

```
84
```

test-image-local2.vsc

```
func foo(int a, bool b) -> int {
    int c;
    c = a + 10;
    return c;
}

func main() -> int {
    bool b = false;
    int x = foo(37, b);
    print(x);
    return 0;
}
```

test-image-local2.out

```
47
```

test-matrix-dim.vsc

```
func main() -> int {
    matrix m = [ 0.0, 0.0; 1.1, 1.1; 2.2, 2.2];
    print(m.rowsize);
    print(m.colsize);
    return 0;
}
```

test-matrix-dim.out

```
3
2
```

test-matrix1.vsc

```
func main() -> int {
  matrix x = [1.0;2.0];
  print(x[0,0]);
  return 0;
}
```

test-matrix1.out

```
1.000
```

test-mod1.vsc

```
func main() -> int {
    int a = 5;
    int b = 3;
    int c = 5 % 3;
    print(c);
    return 0;
}
```

test-mod1.out

```
2
```

test-mod2.vsc

```
func main() -> int {
    int x = 15 % 3;
    int y = 1;
    bool tf = (x != y);
    print(tf);
}
```

test-mod2.out

```
1
```

test-mod3.vsc

```
func main() -> int {
    int x = 12 % 18;
    print(x);
    return 0;
}
```

test-mod3.out

```
12
```

test-print1.vsc

```
func main() -> int {
    int x = 5;
    print(x);
    return 0;
}
```

test-print1.out

```
5
```

test-print2.vsc

```
func main() -> int {
    double d = 2.0;
    string s = "hello";
    bool b = true;
    bool x = false;
    print(d);
    print(s);
    print(b);
    print(x);
    return 0;
}
```

test-print2.out

```
2.000
hello
1
0
```

test-print3.vsc

```
func main() -> int {
    matrix m = [1.0, 2.0; 3.0, 4.0; 5.0, 6.0];
    print(m);
    return 0;
}
```

test-print3.out

```
1.000 2.000
3.000 4.000
5.000 6.000
```

test-save1.vsc

```
func main() -> int {
    matrix m_hard = [ 0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0,
     13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0,
     27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0, 40.0,
     41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                      0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0,
     13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0,
     27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0, 40.0,
     41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                      0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0,
     13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0,
     27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0, 40.0,
     41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                      0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0,
     13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0,
     27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0, 40.0,
     41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                      0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0,
     13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0,
     27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0, 40.0,
     41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
```

```
                  0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0,
13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0,
27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0, 40.0,
41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                  0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0,
13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0,
27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0, 40.0,
41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                  0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0,
13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0,
27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0, 40.0,
41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                  0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0,
13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0,
27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0, 40.0,
41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                  0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0,
13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0,
27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0, 40.0,
41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                  0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0,
13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0,
27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0, 40.0,
41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                  0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0,
13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0,
27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0, 40.0,
41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                                        0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0,
9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0,
23.0, 24.0, 25.0, 26.0, 27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0,
37.0, 38.0, 39.0, 40.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                                        0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0,
9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0,
23.0, 24.0, 25.0, 26.0, 27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0,
37.0, 38.0, 39.0, 40.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                                        0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0,
9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0,
23.0, 24.0, 25.0, 26.0, 27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0,
37.0, 38.0, 39.0, 40.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                                        0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0,
9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0,
23.0, 24.0, 25.0, 26.0, 27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0,
37.0, 38.0, 39.0, 40.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                                        0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0,
9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0,
23.0, 24.0, 25.0, 26.0, 27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0,
37.0, 38.0, 39.0, 40.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                                        0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0,
9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0,
23.0, 24.0, 25.0, 26.0, 27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0,
37.0, 38.0, 39.0, 40.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                                        0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0,
9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0,
23.0, 24.0, 25.0, 26.0, 27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0,
37.0, 38.0, 39.0, 40.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                                        0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0,
9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0,
23.0, 24.0, 25.0, 26.0, 27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0,
37.0, 38.0, 39.0, 40.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                                        0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0,
```

```
9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0,
23.0, 24.0, 25.0, 26.0, 27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0,
37.0, 38.0, 39.0, 40.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                                       0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0,
9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0,
23.0, 24.0, 25.0, 26.0, 27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0,
37.0, 38.0, 39.0, 40.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                                       0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0,
9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0,
23.0, 24.0, 25.0, 26.0, 27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0,
37.0, 38.0, 39.0, 40.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                 0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0,
13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0,
27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0, 40.0,
41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                 0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0,
13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0,
27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0, 40.0,
41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                 0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0,
13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0,
27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0, 40.0,
41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                 0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0,
13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0,
27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0, 40.0,
41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                 0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0,
13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0,
27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0, 40.0,
41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                 0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0,
13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0,
27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0, 40.0,
41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                 0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0,
13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0,
27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0, 40.0,
41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                 0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0,
13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0,
27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0, 40.0,
41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                 0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0,
13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0,
27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0, 40.0,
41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                 0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0,
13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0,
27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0, 40.0,
41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                 0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0,
13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0,
27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0, 40.0,
41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                   0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0,
12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0,
26.0, 27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0,
40.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                   0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0,
12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0,
```

```
   26.0, 27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0,
   40.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                       0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0,
   12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0,
   26.0, 27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0,
   40.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                       0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0,
   12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0,
   26.0, 27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0,
   40.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                       0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0,
   12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0,
   26.0, 27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0,
   40.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                       0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0,
   12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0,
   26.0, 27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0,
   40.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                       0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0,
   12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0,
   26.0, 27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0,
   40.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                       0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0,
   12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0,
   26.0, 27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0,
   40.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                       0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0,
   12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0,
   26.0, 27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0,
   40.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                       0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0,
   12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0,
   26.0, 27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0,
   40.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                       0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0,
   12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0,
   26.0, 27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0,
   40.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                       0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0,
   12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0,
   26.0, 27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0,
   40.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                       0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0,
   12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0,
   26.0, 27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0,
   40.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                       0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0,
   12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0,
   26.0, 27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0,
   40.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                       0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0,
   12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0,
   26.0, 27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0,
   40.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                       0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0,
   12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0,
   26.0, 27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0,
   40.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ;
                       0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0,
   12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0,
   26.0, 27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0,
   40.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0 ];
image im = (m_hard, m_hard, m_hard);
save(im);
  return 0;
```

```
}
```

test-save1.out

test-var1.vsc

```
func main() -> int {
  int a;
  a = 42;
  print(a);
  return 0;
}
```

test-var1.out

```
42
```

test-var2.vsc

```
func foo(int c) -> int {
  return c + 42;
}

func main() -> int {
  int a = foo(73);
  print(a);
  return 0;
}
```

test-var2.out

```
115
```

test-while1.vsc

```
func main() -> int {
  int i;
  i = 5;
  while (i > 0) {
    print(i);
    i = i - 1;
  }
  print(42);
  return 0;
}
```

test-while1.out

```
5
4
3
2
1
```

```
42
```

test-while2.vsc

```
func foo(int a) -> int {
  int j;
  j = 0;
  while (a > 0) {
    j = j + 2;
    a = a - 1;
  }
  return j;
}

func main() -> int {
  print(foo(7));
  return 0;
}
```

test-while2.out

```
14
```

10 Full Code Listing

10.1 parser.mly

```
/*
File: PARSER.MLY
Description: Parses input
*/

%{
open Ast
%}

%token SEMI LPAREN RPAREN LBRACE RBRACE COMMA DOT
%token PLUS MINUS TIMES DIVIDE ASSIGN NOT
%token EQ NEQ LT LEQ GT GEQ TRUE FALSE AND OR
%token RETURN IF ELSE FOR WHILE INT BOOL VOID
%token DOUBLE STRING MATRIX ROWSIZE COLSIZE IMAGE RED GREEN BLUE
%token MOD FUNC ARROW LBRACK RBRACK
%token <int> INT_LITERAL
%token <float> DBL_LITERAL
%token <string> STR_LITERAL
%token <string> ID
%token EOF

%nonassoc NOELSE
%nonassoc ELSE
%right ASSIGN
%left COMMA
%left OR
%left AND
%left EQ NEQ
%left LT GT LEQ GEQ
%left PLUS MINUS
%left TIMES DIVIDE MOD
%right NOT NEG
%left LPAREN RPAREN

%start program
%type <Ast.program> program

%%

program:
  decls EOF { $1 }

decls:
   /* nothing */ { [] }
 | decls fdecl { $2 :: $1 }

fdecl:
  FUNC ID LPAREN formals_opt RPAREN ARROW typ LBRACE stmt_list RBRACE
     { { typ = $7;
     fname = $2;
     formals = $4;
     body = List.rev $9 } }

formals_opt:
    /* nothing */ { [] }
  | formal_list   { List.rev $1 }
```

```
formal_list:
    typ ID                   { [($1,$2)] }
  | formal_list COMMA typ ID { ($3,$4) :: $1 }

typ:
    INT { Int }
  | BOOL { Bool }
  | VOID { Void }
  | DOUBLE { Double }
  | STRING { String }
  | IMAGE { Image }
  | MATRIX { Matrix }

stmt_list:
    /* nothing */  { [] }
  | stmt_list stmt { $2 :: $1 }

stmt:
    expr SEMI { Expr $1 }
  | RETURN SEMI { Return Noexpr }
  | RETURN expr SEMI { Return $2 }
  | LBRACE stmt_list RBRACE { Block(List.rev $2) }
  | IF LPAREN expr RPAREN stmt %prec NOELSE { If($3, $5, Block([])) }
  | IF LPAREN expr RPAREN stmt ELSE stmt    { If($3, $5, $7) }
  | FOR LPAREN expr_opt SEMI expr SEMI expr_opt RPAREN stmt
     { For($3, $5, $7, $9) }
  | WHILE LPAREN expr RPAREN stmt { While($3, $5) }
  | typ ID SEMI { Local($1, $2, Noassign($1)) }
  | typ ID ASSIGN expr SEMI { Local($1, $2, $4) }

expr_opt:
    /* nothing */ { Noexpr }
  | expr          { $1 }

expr:
    INT_LITERAL      { IntLit($1) }
  | DBL_LITERAL      { DblLit($1) }
  | STR_LITERAL      { StrLit($1) }
  | TRUE             { BoolLit(true) }
  | FALSE            { BoolLit(false) }
  | ID               { Id($1) }
  | matrix_literal   { MatLit($1) }
  | LPAREN expr RPAREN { $2 }
  | expr PLUS   expr { Binop($1, Add,   $3) } /* when writing binop in .ml file, need to
    overload op */
  | expr MINUS  expr { Binop($1, Sub,   $3) }
  | expr TIMES  expr { Binop($1, Mult,  $3) }
  | expr DIVIDE expr { Binop($1, Div,   $3) }
  | expr EQ     expr { Binop($1, Equal, $3) }
  | expr NEQ    expr { Binop($1, Neq,   $3) }
  | expr MOD    expr { Binop($1, Mod,   $3) }
  | expr LT     expr { Binop($1, Less,  $3) }
  | expr LEQ    expr { Binop($1, Leq,   $3) }
  | expr GT     expr { Binop($1, Greater, $3) }
  | expr GEQ    expr { Binop($1, Geq,   $3) }
  | expr AND    expr { Binop($1, And,   $3) }
  | expr OR     expr { Binop($1, Or,    $3) }
  | MINUS expr %prec NEG { Unop(Neg, $2) }
  | NOT expr         { Unop(Not, $2) }
  | ID ASSIGN expr   { Assign($1, $3) }
```

```
  | func_id LPAREN actuals_opt RPAREN { Call($1, $3) }
  | LPAREN ID COMMA ID COMMA ID RPAREN { ImageLit($2, $4, $6) }
  | ID DOT RED  { ImageRedAccess($1) }
  | ID DOT GREEN  { ImageGreenAccess($1) }
  | ID DOT BLUE  { ImageBlueAccess($1) }
  | ID DOT ROWSIZE { MatrixRowSize($1) }
  | ID DOT COLSIZE { MatrixColSize($1) }
  | ID LBRACK expr COMMA expr RBRACK { MatAccess($1, $3, $5) }
  | LPAREN typ RPAREN expr             { Cast ($2, $4) }

func_id:
    ID                  {$1}

actuals_opt:
    /* nothing */ { [] }
  | actuals_list  { List.rev $1 }

actuals_list:
    expr                    { [$1] }
  | actuals_list COMMA expr { $3 :: $1 }

matrix_literal:
   LBRACK RBRACK                { [[]] } /* Empty matrix */
 | LBRACK matrix_body RBRACK    { $2 }

matrix_body:
    matrix_row                   { [$1] }
  | matrix_body SEMI matrix_row  { $3 :: $1 }

matrix_row:
   DBL_LITERAL                                          { [DblLit($1)] }
  | MINUS DBL_LITERAL %prec NEG                         { [Unop(Neg, DblLit($2))] }
  | matrix_row COMMA DBL_LITERAL                        { DblLit($3) :: $1 }
  | matrix_row COMMA MINUS DBL_LITERAL %prec NEG    { Unop(Neg, DblLit($4)) :: $1 }
```

10.2 scanner.mll

```
(*
File: SCANNER.MLL
Description: Scans input
*)

{ open Parser }

rule token = parse
  [' ' '\t' '\r' '\n'] { token lexbuf } (* Whitespace *)
| "/*"     { multicomment lexbuf }      (* Comments *)
| "//"     { singlecomment lexbuf}
| '('      { LPAREN }
| ')'      { RPAREN }
| '{'      { LBRACE }
| '}'      { RBRACE }
| ';'      { SEMI }
| ','      { COMMA }
| '.'        { DOT }
| '+'      { PLUS }
| '-'      { MINUS }
| '*'      { TIMES }
```

```
| '/'      { DIVIDE }
| '='      { ASSIGN }
| "=="     { EQ }
| "!="     { NEQ }
| '<'      { LT }
| "<="     { LEQ }
| ">"      { GT }
| ">="     { GEQ }
| "&&"     { AND }
| "||"     { OR }
| "!"      { NOT }
| "if"     { IF }
| "else"   { ELSE }
| "for"    { FOR }
| "while"  { WHILE }
| "return" { RETURN }
| "int"    { INT }
| "bool"   { BOOL }
| "double" { DOUBLE }
| "string" { STRING }
| "matrix" { MATRIX }
| "image"  { IMAGE }
| "red"    { RED }
| "green"  { GREEN }
| "blue"   { BLUE }
| "rowsize" { ROWSIZE }
| "colsize" { COLSIZE }
| "void"   { VOID }
| "true"   { TRUE }
| "false"  { FALSE }
| '['        { LBRACK }
| ']'        { RBRACK }
| "func"   { FUNC }
| '%'      { MOD }
| "->"     { ARROW }

| ['0'-'9']+ as lxm { INT_LITERAL(int_of_string lxm) }
| ['0'-'9']*'.'['0'-'9']+ as lxm { DBL_LITERAL(float_of_string lxm) }
| ['a'-'z' 'A'-'Z']['a'-'z' 'A'-'Z' '0'-'9' '_']* as lxm { ID(lxm) }
| '"' ([^ '"']* as lxm) '"' { STR_LITERAL(lxm) }
| eof { EOF }
| _ as char { raise (Failure("illegal character " ^ Char.escaped char)) }

and singlecomment = parse
  "\n" { token lexbuf }
| _    { singlecomment lexbuf }

and multicomment = parse
  "*/" { token lexbuf }
| _    { multicomment lexbuf }
```

10.3 ast.ml

```
(*
File: AST.ML
Description: Creates AST from parser
*)
type op = Add | Sub | Mult | Div | Equal | Neq | Less | Leq | Greater | Geq |
```

```
          And | Or | Mod

type uop = Neg | Not


type typ = Int | Bool | Void | Double | String | Image
          | Matrix | DimMatrix of int * int

type bind = typ * string

type expr =
    IntLit of int
  | StrLit of string
  | DblLit of float
  | BoolLit of bool
  | Id of string
  | Binop of expr * op * expr
  | Unop of uop * expr
  | Assign of string * expr
  | Call of string * expr list
  | Noexpr
  | Noassign of typ
  | MatLit of expr list list
  | MatAccess of string * expr * expr
  | ImageLit of string * string * string
  | ImageRedAccess of string (* Can only imageaccess on a image id *)
  | ImageGreenAccess of string
  | ImageBlueAccess of string
  | MatrixRowSize of string
  | MatrixColSize of string
  | Cast of typ * expr

type stmt =
    Block of stmt list
  | Expr of expr
  | Return of expr
  | If of expr * stmt * stmt
  | For of expr * expr * expr * stmt
  | While of expr * stmt
  | Local of typ * string * expr

type func_decl = {
    typ : typ;
    fname : string;
    formals : bind list;
    body : stmt list;
  }

type program = func_decl list

(* Pretty-printing functions *)

let string_of_op = function
    Add -> "+"
  | Sub -> "-"
  | Mult -> "*"
  | Div -> "/"
  | Equal -> "=="
  | Neq -> "!="
  | Less -> "<"
  | Leq -> "<="
```

```
  | Greater -> ">"
  | Geq -> ">="
  | And -> "&&"
  | Or -> "||"
  | Mod -> "%"


let string_of_uop = function
    Neg -> "-"
  | Not -> "!"

let string_of_typ = function
    Int -> "int"
  | Bool -> "bool"
  | Void -> "void"
  | Double -> "double"
  | String -> "string"
  | Image -> "image"
  | Matrix -> "matrix"
  | DimMatrix(l1, l2)-> "matrix" ^ "[" ^ string_of_int l1 ^ ", " ^ string_of_int l2 ^ "]"

let rec string_of_expr = function
    IntLit(l) -> string_of_int l
  | StrLit(s) -> s
  | DblLit(l) -> string_of_float l
  | BoolLit(true) -> "true"
  | BoolLit(false) -> "false"
  | Id(s) -> s
  | Binop(e1, o, e2) ->
      string_of_expr e1 ^ " " ^ string_of_op o ^ " " ^ string_of_expr e2
  | Unop(o, e) -> string_of_uop o ^ string_of_expr e
  | Assign(v, e) -> v ^ " = " ^ string_of_expr e
  | Call(f, el) ->
      f ^ "(" ^ String.concat ", " (List.map string_of_expr el) ^ ")"
  | Noexpr -> ""
  | Noassign(t) -> string_of_typ t
  | MatLit(ell) -> (* TEST *)
      "[" ^ String.concat "; "
      (List.map (fun d -> String.concat ", " (List.map (* string_of_float *)
     string_of_expr d)) ell) ^ "]"
  | MatAccess(s, l1, l2) -> s ^ "[" ^ string_of_expr l1 ^ ", " ^ string_of_expr l2 ^ "]"
  | ImageLit(m1, m2, m3) -> "image(" ^ m1 ^ ", " ^ m2 ^ ", " ^ m3 ^ ")"
  | ImageRedAccess(s) -> s ^ ".red"
  | ImageGreenAccess(s) -> s ^ ".green"
  | ImageBlueAccess(s) ->  s ^ ".blue"
  | MatrixRowSize(s) -> s ^ ".rowsize"
  | MatrixColSize(s) -> s ^ ".colsize"
  | Cast(t, e) -> "(" ^ string_of_typ t ^ ")" ^ string_of_expr e


let rec string_of_stmt = function
    Block(stmts) ->
      "{\n" ^ String.concat "" (List.map string_of_stmt stmts) ^ "}\n"
  | Expr(expr) -> string_of_expr expr ^ ";\n";
  | Return(expr) -> "return " ^ string_of_expr expr ^ ";\n";
  | If(e, s, Block([])) -> "if (" ^ string_of_expr e ^ ")\n" ^ string_of_stmt s
  | If(e, s1, s2) ->  "if (" ^ string_of_expr e ^ ")\n" ^
      string_of_stmt s1 ^ "else\n" ^ string_of_stmt s2
  | For(e1, e2, e3, s) ->
      "for (" ^ string_of_expr e1  ^ " ; " ^ string_of_expr e2 ^ " ; " ^
      string_of_expr e3  ^ ") " ^ string_of_stmt s
```

```
  | While(e, s) -> "while (" ^ string_of_expr e ^ ") " ^ string_of_stmt s
  | Local(t, s, e) -> if string_of_expr e = ""
      then string_of_typ t ^ " " ^ s ^ ";\n"(* Noassign case *)
      else string_of_typ t ^ " " ^ s ^ " = " ^ string_of_expr e ^ ";\n" (* Local assign
    case *)

let string_of_fdecl fdecl =
  "func " ^ fdecl.fname ^ "(" ^ String.concat ", " (List.map snd fdecl.formals) ^
  ") -> " ^ string_of_typ fdecl.typ
  ^ "\n{\n" ^
  String.concat "" (List.map string_of_stmt fdecl.body) ^
  "}\n"

let string_of_program funcs =
  String.concat "\n" (List.map string_of_fdecl funcs)
```

10.4 semant.ml

```
(*
File: SEMANT.ML
Description: Semantically checks the AST
*)

open Ast

module StringMap = Map.Make(String)

let check (functions) =
  let image_row_size = 50
  and image_col_size = 50 in

  (* Verify a list of bindings has no void types or duplicate names *)
  let check_binds (kind : string) (binds : bind list) =
    List.iter (function
        (Void, b) -> raise (Failure ("illegal void " ^ kind ^ " " ^ b)) (* can't have
    void args*)
      | _ -> ()) binds;                                              (* when iterating
    over list of binds*)
    let rec dups = function
        [] -> ()
      | ((_,n1) :: (_,n2) :: _) when n1 = n2 ->
          raise (Failure ("duplicate " ^ kind ^ " " ^ n1))
      | _ :: t -> dups t
      (* sorts the binds using compare, then checks for dups*)
      in dups (List.sort (fun (_,a) (_,b) -> compare a b) binds)
  in

  (* Collect function declarations for built-in functions: no bodies *)
  let built_in_decls =
    let add_bind map func_def = StringMap.add func_def.fname func_def map
    in List.fold_left add_bind StringMap.empty [
      { typ = Void; fname = "print"; formals = [(String, "x")]; body = [] };
      { typ = Image; fname = "load"; formals = [(String, "x")]; body = [] };
      { typ = Image; fname = "blur"; formals = [(String, "x")]; body = [] };
      { typ = Image; fname = "grayscale"; formals = [(String, "x")]; body = [] };
      { typ = Image; fname = "brighten"; formals = [(String, "x")]; body = [] };
      { typ = Image; fname = "edgedetect"; formals = [(String, "x")]; body = [] };
      { typ = Matrix; fname = "dim"; formals = [(String, "x")]; body = [] };
```

```
    { typ = Int; fname = "row_size"; formals = [(String, "filename")]; body = [] };
    { typ = Double; fname = "int2double"; formals = [(Int, "i")]; body = [] };
    { typ = Double; fname = "dbl_arr"; formals = []; body = [] };
    { typ = Bool; fname = "save"; formals = [(Image, "x")]; body = [] };
  ]
in

(* Add function name to symbol table *)
let add_func map fd =
  match fd.typ with
    Matrix -> raise (Failure ("Function cannot return matrix type"))
  | Image -> raise (Failure ("Function cannot return image type"))
  | _ ->     let built_in_err = "function " ^ fd.fname ^ " may not be defined"
      and dup_err = "duplicate function " ^ fd.fname
      and make_err er = raise (Failure ("Adding func failed with " ^ er))
      and n = fd.fname (* Name of the function *)
      in match fd with (* No duplicate functions or redefinitions of built-ins *)
           _ when StringMap.mem n built_in_decls -> make_err built_in_err
         | _ when StringMap.mem n map -> make_err dup_err
         | _ ->  StringMap.add n fd map
in

(* Collect all function names into one symbol table *)
let function_decls = List.fold_left add_func built_in_decls functions
in

(* Return a function from our symbol table *)
let find_func s =
  try StringMap.find s function_decls
  with Not_found -> raise (Failure ("unrecognized function " ^ s))
in

(* Ensure "main" is defined *)
let _ = find_func "main" in

let check_function func =
  (* Make sure no formals are void or duplicates *)
  check_binds "formal" func.formals;

  (* Raise exception if the given rvalue type cant be assigned to
     the given LLValue type *)
  let check_assign lvaluet rvaluet err =
     if lvaluet = rvaluet then lvaluet
     else raise (Failure ("Check_assign failed with " ^ err))
  in

  (* Build symbol table of formal variables for this function *)
  let formals = List.fold_left (fun m (ty, name) ->
      StringMap.add name ty m) StringMap.empty ( func.formals )
  in

  (* Return a variable from our local symbol table *)
  let type_of_identifier s symbols =
    try StringMap.find s symbols
    with Not_found -> raise (Failure ("undeclared identifier " ^ s))
  in

  let matrix_access_type = function
      DimMatrix(_, _) -> Double
    | _ -> raise (Failure ("Illegal matrix access"))
  in
```

```
(* Return a semantically-checked expression, i.e., with a type *)
let rec check_expr symbols = function
    IntLit _ -> Int
  | StrLit _ -> String
  | DblLit _  -> Double
  | BoolLit _ -> Bool
  | Binop (e1, op, e2) ->
      let t1 = check_expr symbols e1
      and t2 = check_expr symbols e2
      in let same = (t1 = t2)
      (* check if matrix, then check dimensions are the same *)
      in let t1_dim = (match t1 with
          DimMatrix(m, n) -> (m, n)
        | _ -> (-1, -1) (* for non - matrices *)
      )
      in let t2_dim = (match t2 with
          DimMatrix(m, n) -> (m, n)
        | _ -> (-1,-1)
      )
      in (match op with
          Add | Sub | Mult | Div | Mod when same && t1 = Int -> Int
        | Add | Sub | Mult | Div | Mod when same && t1 = Double -> Double
        | Equal | Neq when same -> Bool
        | Less | Leq | Greater | Geq when same -> Bool
        | And | Or when same && t1 = Bool -> Bool
        (* matrix op matrix *)
        | Add | Sub when t1 = DimMatrix(fst t1_dim, snd t1_dim)
                      && t2 = DimMatrix(fst t2_dim, snd t2_dim) ->
                      if same then DimMatrix(fst t1_dim, snd t1_dim)
                      else raise (Failure ("Cannot add/subtract matrices with
different dimensions"))
        | Mult when t1 = DimMatrix(fst t1_dim, snd t1_dim)
                && t2 = DimMatrix(fst t2_dim, snd t2_dim) ->
                if (snd t1_dim = fst t2_dim) then DimMatrix(fst t1_dim, snd t2_dim)
                else raise (Failure ("Matrices cannot be multiplied given their
dimensions"))
        (* matrix op scalar *)
        | Add | Sub | Mult when not same
                              && t1 = DimMatrix(fst t1_dim, snd t1_dim) ->
                              if ((t2 = Int) || (t2 = Double))
                                  then DimMatrix(fst t1_dim, snd t1_dim)
                              else raise (Failure ("Scalar ops with matrices can
only use ints or doubles"))
        (* scalar op matrix *)
        | Add | Sub | Mult when not same
                              && t2 = DimMatrix(fst t2_dim, snd t2_dim) ->
                              if ((t1 = Int) || (t1 = Double)) then DimMatrix(fst
t2_dim, snd t2_dim)
                              else raise (Failure ("Scalar ops with matrices can
only use ints or doubles"))
        | _ -> raise (Failure ("illegal binary operator")))
  | Unop (op, e) ->
        let list_acceptable = [Int; Double]
        and typ_e = (check_expr symbols e)
        in if op = Neg then
          if List.mem typ_e list_acceptable then typ_e
          else raise (Failure ("Cannot negate a non-int/double"))
        else
          if typ_e = Bool then Bool (* if op = Not *)
          else raise (Failure ("Cannot NOT a non-bool"))
```

```
  | Assign (varname, e) ->
        let lt = type_of_identifier varname symbols
        and rt = check_expr symbols e
        and non_mat_or_img = [Int; Bool; Double; String] (* Because matrices and
images have both DimImg and DimMat *)
        in if List.mem rt non_mat_or_img (* If the variable is not a matrix then make
sure the two sides are same *)
          then check_assign lt rt "illegal assignment"
          else lt
  | Call (fname, actuals) -> let func_call = find_func fname in (* make sure function
exists *)
        if List.length actuals != List.length func_call.formals
          then raise (Failure ("expecting " ^ string_of_int (List.length
func_call.formals) ^ " arguments in " ^ fname))
        else
            (* Do not conduct type check on argument for "print" method *)
          if not (String.equal fname "print")
            then
              List.iter2
              (fun (ft, _) e ->
                let et = check_expr symbols e
                in ignore (check_assign ft et "illegal actual argument"))
              func_call.formals actuals;
              func_call.typ
  | Noexpr -> Void
  | Noassign (t) -> t
  | MatLit m -> (* type checking done in parser/scanner *)
        let rec len_check = function
            []                  -> true (* empty list of lists *)
          | _ :: []             -> true (* only one list *)
          (* two lists, check their lengths *)
          | fst :: snd :: []    -> List.length fst = List.length snd
          (* more than two lists, recursively compare first with second & second with
rest *)
          | fst :: snd :: tail  -> len_check (fst::[snd]) && len_check (snd::tail)
        in if len_check m then
          let rows = (List.length m) and cols = (List.length (List.hd m)) in
          (* if true then raise (Failure ((string_of_int rows) ^ " by " ^
(string_of_int cols))) *)
          (* it registers empty matrix as 1 by 0? *)
          if rows = 1 && cols = 0 then DimMatrix(0, 0)
          else DimMatrix(rows, cols)
        else raise (Failure ("Not all rows in matrix are the same length"))
  | MatAccess (mname, row, col) ->
        let _ = (match (check_expr symbols row) with
            Int -> Int
          | _ -> raise (Failure ("Attempting to access with a non-integer type")))
        and _ = (match (check_expr symbols col) with
            Int -> Int
          | _ -> raise (Failure ("Attempting to access with a non-integer type")))
        in matrix_access_type (type_of_identifier mname symbols)
  | Cast (typ, s1) ->
        let typ_s1 = (check_expr symbols s1) in
        if typ = Int
          then if typ_s1 = Double then Int
          else raise (Failure ("Cannot cast non-double to int"))
        else if typ = Double
          then if typ_s1 = Int then Double
          else raise (Failure ("Cannot cast non-int to a double"))
        else raise (Failure("Cannot cast to that type"))
  | Id varname ->
```

```
        if StringMap.mem varname symbols then type_of_identifier varname symbols
        else raise (Failure ("Variable not found: " ^ varname))
  | ImageLit (m1, m2, m3) ->  (* need to check that all matrices are same size*)
      let m1t = (type_of_identifier m1 symbols)
         and m2t = (type_of_identifier m2 symbols)
         and m3t = (type_of_identifier m3 symbols)
         in
         if m1t = m2t
           then if m2t = m3t
             then if m3t = DimMatrix(image_row_size, image_col_size)
               then Image
             else raise(Failure ("Can't create an image with the wrong dimensions"))
           else raise(Failure ("Matrix sizes don't match"))
         else raise(Failure ("Matrix sizes don't match"))
  | ImageRedAccess varname ->
        if StringMap.mem varname symbols
          then if (type_of_identifier varname symbols) = Image then Matrix
          else raise (Failure ("Cannot call .red on non image datatype"))
        else raise (Failure ("Image variable does not exist"))
  | ImageGreenAccess (varname) ->
        if StringMap.mem varname symbols
          then if (type_of_identifier varname symbols) = Image then Matrix
          else raise (Failure ("Cannot call .green on non image datatype"))
        else raise (Failure ("Image variable does not exist"))
  | ImageBlueAccess (varname) ->
        if StringMap.mem varname symbols
          then if (type_of_identifier varname symbols) = Image then Matrix
          else raise (Failure ("Cannot call .blue on non image datatype"))
        else raise (Failure ("Image variable does not exist"))
  | MatrixRowSize (varname) ->
        if StringMap.mem varname symbols
          then if (type_of_identifier varname symbols) = Matrix then Int
          else raise (Failure ("Cannot call .rowsize on non matrix datatype"))
        else raise (Failure ("Matrix variable does not exist"))
  | MatrixColSize (varname) ->
        if StringMap.mem varname symbols
          then if (type_of_identifier varname symbols) = Matrix then Int
          else raise (Failure ("Cannot call .colsize on non matrix catatype"))
        else raise (Failure ("Matrix variable does not exist"))
in

let check_bool_check_expr symbols e = if check_expr symbols e != Bool
    then raise (Failure ("expected Boolean expression")) else ()
in

(* Return a semantically-checked statement i.e. containing sexprs *)
let rec check_stmt symbols = function
    Expr e -> ignore (check_expr symbols e) ; symbols
  | If(cond, b1, b2) ->
        check_bool_check_expr symbols cond;
        ignore(check_stmt symbols b1); check_stmt symbols b2
  | For(e1, cond, e2, st) ->
        ignore (check_expr symbols e1) ;
        check_bool_check_expr symbols cond ;
        ignore (check_expr symbols e2) ;
        check_stmt symbols st
  | While(cond, st) -> check_bool_check_expr symbols cond; check_stmt symbols st
  | Return e ->
        let t = check_expr symbols e
        in (match t with
            DimMatrix(_, _) -> raise (Failure ("Cannot return matrix type (don't know
```

```
    how to allocate result)"))
              | _ -> if t = func.typ then symbols else raise (Failure ("Return type does
    not match method signature")))
      | Block sl ->
            let rec check_stmt_list symbols_rec = function
                [Return _ as s] -> check_stmt symbols_rec s
              | Return _ :: _   -> raise (Failure "nothing may follow a return")
              | Block sl :: ss  -> check_stmt_list symbols_rec (sl @ ss) (* Flatten
    blocks *)
              | s :: ss         -> let symbols_updated = check_stmt symbols_rec s
                                   in check_stmt_list symbols_updated ss
              | []              -> symbols_rec
            in check_stmt_list symbols sl
      | Local (typ, varname, e) as call ->
            if StringMap.mem varname symbols then
              (ignore (check_expr symbols e) ; symbols)
            else
              let expr_type = check_expr symbols e
              in if expr_type = Void then
                check_stmt (StringMap.add varname expr_type symbols) call
              else if typ = Matrix then
                check_stmt (StringMap.add varname expr_type symbols) call (* Adds the
    dimmat *)
              else if typ = expr_type then
                check_stmt (StringMap.add varname expr_type symbols) call
              else raise (Failure ("Local var type does not match"))
  in check_stmt formals (Block func.body)
in (List.map check_function functions)
```

10.5 codegen.ml

```
(*
File: CODEGEN.ML
Description: Generates LLVM IR
*)

module L = Llvm
module I = Int64
open Ast

module StringMap = Map.Make(String)


(* SECTION 1: ast.program -> Llvm.module *)
let translate functions =

  let image_row_size = 50
  and image_col_size = 50 in

(* SECTION 2: TYPES *)
(* SECTION 2a: Defining LLVM Types *)
  let context      = L.global_context ()
  in let the_module   = L.create_module context "VSCOde"
  and double_t     = L.double_type  context
  and i32_t        = L.i32_type     context
  and i8_t         = L.i8_type      context
  and i1_t         = L.i1_type      context
  and void_t       = L.void_type    context
```

```
  and array_t      = L.array_type
  in let str_t        = L.pointer_type i8_t
  and print_t      = L.var_arg_function_type i32_t [| L.pointer_type i8_t |]
  in let image_t      = L.named_struct_type context "image_t"
    in L.struct_set_body image_t [| (array_t (array_t double_t image_col_size)
    image_row_size );
                                    (array_t (array_t double_t image_col_size)
    image_row_size );
                                    (array_t (array_t double_t image_col_size)
    image_row_size ) |] false;


(* SECTION 2b: AST typ to LLVM typ conversion *)
  let ltype_of_typ = function
      Int               -> i32_t
    | Bool              -> i1_t
    | Void              -> void_t
    | Double            -> double_t
    | String            -> str_t
    | Matrix            -> void_t (* this should never be called because we calculate
    matrix size and match DimMatrix. Is void_t ok?*)
    | DimMatrix (r, c)  -> (array_t (array_t double_t c) r )
    | Image   ->           image_t
  in


(* SECTION 3: FUNCTION/BLOCKS *)
(* SECTION 3a: Making map of all of the user functions
key: function name
val: tuple: 1. ll.value which can be a function, global var, constant, etc
            2. func decl *)

    (* Built our built in functions and add then to build_in_function_decls,
    which is used as the starting point for function_decls *)
    let built_in_function_decls : (L.llvalue * func_decl) StringMap.t =

      (* Gets the row size (in pixels) of a image *)
      let row_size_cpp_t = L.function_type (i32_t) [| str_t |]
      in let row_size_cpp_func = L.declare_function "row_size_cpp" row_size_cpp_t
    the_module
      in let row_size_func_decl =
        { typ = Int;
          fname = "row_size";
          formals = [(String, "filename")];
          body = [] }
      in let row_size_func = L.define_function "row_size" (L.function_type i32_t [| str_t
    |]) the_module

      in let row_size_func_body function_ptr =
        let builder = ref (L.builder_at_end context (L.entry_block function_ptr))
        in let path = List.hd (Array.to_list (L.params function_ptr))
        in let row_size_return = L.build_call row_size_cpp_func [| path |] "row_size_ret"
    !builder
        in ignore(L.build_ret row_size_return !builder)
      in ignore (row_size_func_body row_size_func);

      StringMap.add "row_size" (row_size_func, row_size_func_decl) StringMap.empty
      in


 (* Now add all the user defined functions to function_decls *)
```

```
  let function_decls : (L.llvalue * func_decl) StringMap.t =
    let function_decl m fdecl =
      let name = fdecl.fname
      and formal_types = Array.of_list (List.map (fun (t,_) -> ltype_of_typ t)
    fdecl.formals)
      in let ftype = L.function_type (ltype_of_typ fdecl.typ) formal_types
      in StringMap.add name (L.define_function name ftype the_module, fdecl) m
    in List.fold_left function_decl built_in_function_decls functions
  in


(* SECTION 3b: Adds a return statement to the end of every block *)
  let add_terminal builder_and_3maps_tuple instr =
    match L.block_terminator (L.insertion_block (fst builder_and_3maps_tuple)) with
        Some _ -> ()
      | None -> ignore (instr (fst builder_and_3maps_tuple))
  in

(* SECTION 4: PRINT *)
(* SECTION 4a: Formating strings for printing *)
  let str_format_str builder = L.build_global_stringptr "%s\n" "fmt" builder
  in let int_format_str builder = L.build_global_stringptr "%d\n" "fmt" builder
  in let double_format_str builder = L.build_global_stringptr "%.3f\n" "fmt" builder
  in let mat_format_str builder row column =
    let make_str r c =
      let rec row str orig_c r' c' =
        if (r' = 0) then str
        else
            if (c' = 0) then let new_str  = str ^ "\n"
              in row new_str orig_c (r'-1) orig_c
            else let new_str = str ^ "%.3f "
              in row new_str orig_c r' (c'-1)
      in row "" c r c
    in let mat_str = make_str row column
  in L.build_global_stringptr mat_str "fmt" builder
  in

(* SECTION 4b: Calling C's print function *)
let print_func : L.llvalue =
  L.declare_function "printf" print_t the_module
  in let print_by_type ltyp e' builder mat_dim_map img_dim_map m_row m_col =
    (match ltyp with
        "i32" ->
          ((L.build_call print_func [| int_format_str builder ; e' |]
          "printf" builder), (mat_dim_map, img_dim_map))
      | "i1" -> (* prints 1 for true, 0 for false *)
          ((L.build_call print_func [| int_format_str builder ; e' |]
          "printf" builder), (mat_dim_map, img_dim_map))
      | "double" ->
          ((L.build_call print_func [| double_format_str builder ; e' |]
          "printf" builder), (mat_dim_map, img_dim_map))
      | "i8*" ->  (* string *)
          ((L.build_call print_func [| str_format_str builder ; e' |]
          "printf" builder), (mat_dim_map, img_dim_map))
      | "void" -> raise (Failure ("Can't print void type"))
      | _ ->  (* must be matrix *)
          ((L.build_call print_func [| (mat_format_str builder m_row m_col) ; e' |]
          "printf" builder), (mat_dim_map, img_dim_map)))
  in

(* SECTION 5: MATRICES *)
```

```
(* SECTION 5a: General Matrix Helper Functions*)
  (* SECTION 5a i: Checks if an id is a matrix *)
    let check_if_matrix e_typ =
      (match e_typ with
          "i32" -> false
        | "i1" -> false
        | "void" -> false
        | "double" -> false
        | "i8*" -> false (* string *)
        | _ -> true (* now it should be a matrix... *)
      )

  (* SECTION 5a ii: Gets Matrix Dimensions *)
    and get_matrix_dim m =
      let rec len_check = function
          []                  -> true (* empty list of lists *)
        | _ :: []             -> true (* only one list *)
        | fst :: snd :: []    -> List.length fst = List.length snd
        | fst :: snd :: tail  -> len_check (fst::[snd]) && len_check (snd::tail)
            in if len_check m
                then let rows = (List.length m)
                and cols = (List.length (List.hd m))
                  in if rows = 1 && cols = 0 then (0,0) (* An "empty" matrix has 1 row
    (empty) *)
                  else (rows, cols)
            else raise (Failure ("Not all rows in matrix are the same length"))

  (* SECTION 5a iii: Accesses element at given i, j position *)
    and build_matrix_access m row_index column_index builder locals_map mat_dim_map =
     (try let value = StringMap.find m locals_map
       in (let (r, c) = StringMap.find m mat_dim_map
        in if L.int64_of_const row_index < Some (I.of_int r)
          && L.int64_of_const column_index < Some (I.of_int c)
          && L.int64_of_const row_index >= Some (I.of_int 0)
          && L.int64_of_const column_index >= Some (I.of_int 0)
        then L.build_load (L.build_gep (value) [| L.const_int i32_t 0; row_index;
            column_index |] m builder) m builder
        else raise (Failure ("Index out of matrix bounds")))
      with Not_found -> raise (Failure ("Variable not found 7: " ^ m)))
    in


(* SECTION 5c: Matrix Binop (called within expr) *)
  let build_binop_op op = (match op with
      Add -> L.build_fadd
    | Sub -> L.build_fsub
    | Mult -> L.build_fmul
    | _ -> raise (Failure "Invalid matrix binop"))
  in

  (* SECTION 5c i: Matrix * Matrix add/subtract *)
    let binop_mat_sum op builder mat_dim_map img_dim_map locals_map m1 m2 m1_row m1_col =
      let new_mat_dim_map = StringMap.add "binop_result" (m1_row, m1_col) mat_dim_map
      and new_mat = L.build_alloca (ltype_of_typ (DimMatrix(m1_row, m1_col)))
     "binop_result" builder
      in for i=0 to (m1_row - 1) do
        for j=0 to (m1_col - 1) do
          let elem1' = build_matrix_access m1 (L.const_int i32_t i) (L.const_int i32_t j)
              builder locals_map new_mat_dim_map
          and elem2' = build_matrix_access m2 (L.const_int i32_t i) (L.const_int i32_t j)
              builder locals_map new_mat_dim_map
```

```
        in let result_v = (build_binop_op op) elem1' elem2' "tmp" builder
        in let result_p = L.build_gep new_mat [| L.const_int i32_t 0;
            L.const_int i32_t i; L.const_int i32_t j |] "" builder
        in ignore(L.build_store result_v result_p builder);
      done
    done;
    ((L.build_load (L.build_gep new_mat [| L.const_int i32_t 0 |] "binop_result"
  builder)
        "binop_result" builder), (new_mat_dim_map, img_dim_map))

(* SECTION 5c ii: Matrix * Matrix mult *)
  and binop_mat_mult builder mat_dim_map img_dim_map locals_map m1 m2 m1_row m1_col
  m2_col =
    let new_mat_dim_map = StringMap.add "binop_result" (m1_row, m2_col) mat_dim_map
    and new_mat = L.build_alloca (ltype_of_typ (DimMatrix(m1_row, m2_col)))
  "binop_result" builder
    and tmp_product = L.build_alloca double_t "tmpproduct" builder
    in ignore(L.build_store (L.const_float double_t 0.0) tmp_product builder);
    for i=0 to (m1_row-1) do
      for j=0 to (m2_col-1) do
        ignore(L.build_store (L.const_float double_t 0.0) tmp_product builder);
        for k=0 to (m1_col-1) do
          let m1_float_val = build_matrix_access m1 (L.const_int i32_t i)
              (L.const_int i32_t k) builder locals_map new_mat_dim_map
          and m2_float_val = build_matrix_access m2 (L.const_int i32_t k)
              (L.const_int i32_t j) builder locals_map new_mat_dim_map
          in let product_m1_m2 = L.build_fmul m1_float_val m2_float_val "tmp" builder
          in ignore(L.build_store ( L.build_fadd product_m1_m2
              (L.build_load tmp_product "addtmp" builder) "tmp" builder) tmp_product
  builder);
        done;
        let new_mat_element = L.build_gep new_mat [| L.const_int i32_t 0;
            L.const_int i32_t i; L.const_int i32_t j |] "tmpmat" builder
        in let tmp_product_val = L.build_load tmp_product "resulttmp" builder
        in ignore(L.build_store tmp_product_val new_mat_element builder);
      done
    done;
    ((L.build_load (L.build_gep new_mat [| L.const_int i32_t 0 |] "binop_result"
  builder)
        "binop_result" builder), (new_mat_dim_map, img_dim_map))
    in


(* SECTION 5c iii: Matrix * Scalar mult *)
  let conduct_scalar_mult op builder m1 num mat_dim_map img_dim_map locals_map =
    let (m1_row, m1_col) = StringMap.find m1 mat_dim_map
        in (match op with
             Mult ->
                let new_mat_dim_map = StringMap.add "binop_result" (m1_row, m1_col)
  mat_dim_map
                in let new_mat = L.build_alloca (ltype_of_typ (DimMatrix(m1_row,
  m1_col)))
                    "binop_result" builder
                in for i=0 to (m1_row - 1) do
                  for j=0 to (m1_col - 1) do
                    let elem1' = build_matrix_access m1 (L.const_int i32_t i)
                        (L.const_int i32_t j) builder locals_map new_mat_dim_map
                    in let result_v = (build_binop_op op) elem1' num "tmp" builder
                    in let result_p = L.build_gep new_mat [| L.const_int i32_t 0;
                        L.const_int i32_t i; L.const_int i32_t j |] "" builder
                    in ignore(L.build_store result_v result_p builder);
```

```
                done
              done;
              ((L.build_load (L.build_gep new_mat [| L.const_int i32_t 0 |]
                    "binop_result" builder) "binop_result" builder),
  (new_mat_dim_map, img_dim_map))
          | _ -> raise (Failure "Cannot perform scalar/matrix operation other than
  multiplication")
      ) in

  let binop_dbl_mat_mult op builder mat_dim_map img_dim_map locals_map m1 double_k =
    let num = L.const_float double_t double_k
    in conduct_scalar_mult op builder m1 num mat_dim_map img_dim_map locals_map
  and binop_dblid_mat_mult op builder mat_dim_map img_dim_map locals_map m1 double_id =
    let find_lvalue_of_id name =
      (try let value = StringMap.find name locals_map
        in L.build_load value name builder
      with Not_found -> raise (Failure ("Variable not found 2: " ^ name)))
    in let num = find_lvalue_of_id double_id
    in conduct_scalar_mult op builder m1 num mat_dim_map img_dim_map locals_map

(* SECTION 5d iv: Loading an Image *)
  in let load_image name load_return mat_dim_map img_dim_map locals_map builder =
    let img = L.build_alloca (ltype_of_typ (Image)) "img" builder
    in let r_ptr = L.build_struct_gep img 0 "r_ptr" builder
    and g_ptr = L.build_struct_gep img 1 "g_ptr" builder
    and b_ptr = L.build_struct_gep img 2 "b_ptr" builder

    in let rec turn_llval_to_list as_llval as_list iter =
      let size = 3 * image_row_size * image_col_size
      in (match iter with
        | n when n = size -> as_list (* if it reaches image size then return *)
        | _ -> let next_element = (L.build_load (L.build_gep load_return
              [|L.const_int i32_t iter|] "element_ptr" builder) "element" builder)
              in
              (* FILL IN MATRICES *)
                let mat_list_ix = (iter / 3)
                in let curr_row = (mat_list_ix / image_col_size)
                in let curr_col = (mat_list_ix mod image_col_size)
                  in if iter mod 3 = 2
                    then (* FILL IN RED MATRIX *) let red_elem_ptr =
                        L.build_gep r_ptr [| L.const_int i32_t 0;
                        L.const_int i32_t curr_row; L.const_int i32_t curr_col |]
  "red_mat_ptr" builder
                        in ignore(L.build_store next_element red_elem_ptr builder);
                  else if iter mod 3 = 1
                    then (* FILL IN GREEN MATRIX *) let green_elem_ptr =
                        L.build_gep g_ptr [| L.const_int i32_t 0;
                        L.const_int i32_t curr_row; L.const_int i32_t curr_col |]
  "green_mat_ptr" builder
                    in ignore(L.build_store next_element green_elem_ptr builder);
                  else if iter mod 3 = 0
                    then (* FILL IN BLUE MATRIX *) let blue_elem_ptr =
                        L.build_gep b_ptr [| L.const_int i32_t 0;
                        L.const_int i32_t curr_row; L.const_int i32_t curr_col |]
  "green_mat_ptr" builder
                    in ignore(L.build_store next_element blue_elem_ptr builder);
                  else raise (Failure "Internal error"); (* END ENTIRE IF BLOCK *)
                let new_as_list = as_list@[next_element]
                in (turn_llval_to_list as_llval new_as_list (iter + 1))
      )
      in ignore(turn_llval_to_list load_return [] 0);
```

```
        let new_img_dim_map = StringMap.add name (image_row_size, image_col_size)
    img_dim_map
        in let new_locals_map = StringMap.add name img locals_map

        in (builder, (new_locals_map, (mat_dim_map, new_img_dim_map)))
    in


(* SECTION 6: BINOP EXPR *)
  let binop_scalar typs op e1' e2' builder mat_dim_map img_dim_map =
    (match typs with
        ("double", "double") -> (* 1. DOUBLE *)
          (((match op with
              Add         -> L.build_fadd
            | Sub         -> L.build_fsub
            | Mult        -> L.build_fmul
            | Div         -> L.build_fdiv
            | Equal       -> L.build_fcmp L.Fcmp.Oeq
            | Neq         -> L.build_fcmp L.Fcmp.One
            | Less        -> L.build_fcmp L.Fcmp.Olt
            | Leq         -> L.build_fcmp L.Fcmp.Ole
            | Greater     -> L.build_fcmp L.Fcmp.Ogt
            | Geq         -> L.build_fcmp L.Fcmp.Oge
            | _           -> raise (Failure "internal error: semant should have rejected
    invalid op on double")
          ) e1' e2' "tmp" builder), (mat_dim_map, img_dim_map))
      | ("i32", "i32") ->     (* 2. INT *)
        (((match op with
              Add         -> L.build_add
            | Sub         -> L.build_sub
            | Mult        -> L.build_mul
            | Div         -> L.build_sdiv
            | Mod         -> L.build_srem
            | Equal       -> L.build_icmp L.Icmp.Eq
            | Neq         -> L.build_icmp L.Icmp.Ne
            | Less        -> L.build_icmp L.Icmp.Slt
            | Leq         -> L.build_icmp L.Icmp.Sle
            | Greater     -> L.build_icmp L.Icmp.Sgt
            | Geq         -> L.build_icmp L.Icmp.Sge
            | _           -> raise (Failure "internal error: semant should have rejected
    invalid op on int")
          ) e1' e2' "tmp" builder), (mat_dim_map, img_dim_map))
      | ("i1", "i1") ->     (* 3. BOOL *)
        (((match op with
              And         -> L.build_and
            | Or          -> L.build_or
            | Equal       -> L.build_icmp L.Icmp.Eq
            | Neq         -> L.build_icmp L.Icmp.Ne
            | _           -> raise (Failure "internal error: semant should have rejected
    invalid op on bool")
          ) e1' e2' "tmp" builder), (mat_dim_map, img_dim_map))
      | (_, _) -> raise (Failure ("cannot do binop on different types"))
    )
    in


(* SECTION 7: EXPR *)
(* returns tuple of (LL.value, mat_dim_map) *)
  let rec expr locals_map mat_dim_map img_dim_map builder (e : expr) = match e with
     IntLit i                -> (L.const_int i32_t i, (mat_dim_map, img_dim_map))
```

```
| StrLit i               -> (* this should be a string pointer *)
                           ((L.build_global_stringptr i "str" builder),
(mat_dim_map, img_dim_map))
| DblLit d               -> ((L.const_float double_t d), (mat_dim_map, img_dim_map))
| BoolLit b              ->  if b then (L.const_int i1_t 1, (mat_dim_map,
img_dim_map))
                             else (L.const_int i1_t 0, (mat_dim_map, img_dim_map))
| Id name                -> (try let name' = StringMap.find name locals_map
                              in ((L.build_load name' name builder), (mat_dim_map,
img_dim_map))
                           with Not_found -> raise (Failure ("Variable not found 3:
" ^ name)))
| Binop (e1, op, e2)     ->
      (* e1' is the ll value ll const array array*)
      let e1' = fst (expr locals_map mat_dim_map img_dim_map builder e1)
      and e2' = fst (expr locals_map mat_dim_map img_dim_map builder e2)
      in

      let typ1 = L.string_of_lltype (L.type_of e1')
      and typ2 = L.string_of_lltype (L.type_of e2')
      in

      (* BINOP MATRIX CASE 1: ID op ID *)
      if ((check_if_matrix typ1 = true) && (check_if_matrix typ2 = true))
        then (match (e1, e2) with
          (Id m1, Id m2) ->
            let (m1_row, m1_col) = StringMap.find m1 mat_dim_map
            and (m2_row, m2_col) = StringMap.find m2 mat_dim_map
              in (match op with
                  Add | Sub -> if (m1_row, m1_col) = (m2_row, m2_col)
                    then (binop_mat_sum op builder mat_dim_map img_dim_map locals_map
m1 m2 m1_row m1_col)
                    else raise (Failure "Matrices must have same dimensions for binop
addition")
                | Mult -> if m1_col = m2_row
                    then (binop_mat_mult builder mat_dim_map img_dim_map locals_map
m1 m2 m1_row m1_col m2_col)
                    else raise (Failure "Matrices do not have valid dimensions for
binop multiplication")
                | _ -> raise (Failure "Can't divide two matrices")
              )
          | _ -> raise (Failure "only id matrix op id matrix")
         )

      (* BINOP MATRIX CASE 2: Matrix mult scalar *)
      else if (check_if_matrix typ1 = true) then
        (match (e1, e2) with
            (Id m1, DblLit k) -> binop_dbl_mat_mult op builder mat_dim_map
img_dim_map locals_map m1 k
          | (Id m1, Id id) -> binop_dblid_mat_mult op builder mat_dim_map img_dim_map
locals_map m1 id
          | _ -> raise (Failure "No other types allowed for binop scalar
multiplication"))
      else if (check_if_matrix typ2 = true) then
        (match (e1, e2) with
            (DblLit k, Id m1) -> binop_dbl_mat_mult op builder mat_dim_map
img_dim_map locals_map m1 k
          | (Id id, Id m1) -> binop_dblid_mat_mult op builder mat_dim_map img_dim_map
locals_map m1 id
          | _ -> raise (Failure "No other types allowed for binop scalar
multiplication"))
```

```
      (* NON-MATRIX SCALAR BINOP *)
      else
        let typs = (typ1, typ2)
        in binop_scalar typs op e1' e2' builder mat_dim_map img_dim_map
| Unop (op, e)            ->
      let e' = fst (expr locals_map mat_dim_map img_dim_map builder e)
      in let typ = L.string_of_lltype (L.type_of e')
      in (((match op with
        Neg ->
          (match typ with
              "i32" -> L.build_neg
            | "double" -> L.build_fneg
            | _ -> raise (Failure "internal error: semant should have rejected
invalid unop type")
          )
      | Not when typ = "i1" -> L.build_not
      | _ -> raise (Failure "internal error: semant should have rejected illegal
unop")
    ) e' "tmp" builder), (mat_dim_map, img_dim_map))
| Assign (name, e)       ->
      ( match e with
          MatLit m ->   (* x = [1]; Allocate space according to MatLit and
            update the locals_map for name to have the right amount of space*)
              let (row, col) = get_matrix_dim m
              in let local_var = L.build_alloca (ltype_of_typ (DimMatrix(row,
col))) name builder
              in let new_locals_map = StringMap.add name local_var locals_map
              in let e' = fst (expr new_locals_map mat_dim_map img_dim_map builder
e)
              in ignore (L.build_store e' local_var builder);
              (e', (mat_dim_map, img_dim_map))
          | ImageLit (_, _, _) ->
              let local_var = L.build_alloca (ltype_of_typ (Image)) name builder
              in let new_locals_map = StringMap.add name local_var locals_map
              in let e' = fst (expr new_locals_map mat_dim_map img_dim_map builder
e)
              in ignore(L.build_store e' local_var builder);
              (local_var, (mat_dim_map, img_dim_map))
          | Id idName ->
              let name_value = StringMap.find name locals_map (* value is the
llvalue *)
              in let name_typ = L.type_of name_value
              in if name_typ = (ltype_of_typ (DimMatrix (0, 0)))
                then (try let _ = StringMap.find idName locals_map
                  and (row, col) = StringMap.find idName mat_dim_map
                  in let local_var = L.build_alloca (ltype_of_typ (DimMatrix(row,
col))) name builder
                  in let new_locals_map = StringMap.add name local_var locals_map
                  and new_mat_dim_map = StringMap.add name (row, col) mat_dim_map
                  in let matrix_llarray = fst (expr new_locals_map new_mat_dim_map
img_dim_map builder e)
                  in ignore (L.build_store (fst (expr new_locals_map
new_mat_dim_map img_dim_map builder e))
                      local_var builder);
                  (matrix_llarray, (mat_dim_map, img_dim_map))
                with Not_found -> raise (Failure ("Variable not found 4: " ^
idName)))
              else
                let value = StringMap.find name locals_map
                in let e' = fst (expr locals_map mat_dim_map img_dim_map builder e)
```

```
                in ignore(L.build_store e' value builder);
                (e', (mat_dim_map, img_dim_map))
          | _ -> (* Non matrix type assign *)
                let value = StringMap.find name locals_map (* value is the llvalue *)
                in let e' = fst (expr locals_map mat_dim_map img_dim_map builder e)
                in ignore(L.build_store e' value builder);
                (e', (mat_dim_map, img_dim_map)) (* store v inside e' *)
      )
| Noexpr                -> (L.const_int i32_t 0, (mat_dim_map, img_dim_map))
| Noassign (_)          -> (L.const_int i32_t 0, (mat_dim_map, img_dim_map))
                           (* Placeholder for when something gets assigned to it *)
| MatLit m              ->
      let flipped     = List.map List.rev m
      in let lists       = List.map (List.map (fun e -> fst (expr locals_map
                           mat_dim_map img_dim_map builder e))) flipped
      in let listArray   = List.map Array.of_list lists
      in let listArray2  = List.rev (List.map (L.const_array double_t) listArray)
      in let arrayOfArray  = Array.of_list listArray2
      in ((L.const_array (array_t double_t (List.length (List.hd m))) arrayOfArray),
(mat_dim_map, img_dim_map))
| MatAccess (m, r, c)    ->
      let r' = fst (expr locals_map mat_dim_map img_dim_map builder r)
      and c' = fst (expr locals_map mat_dim_map img_dim_map builder c)
      in ((build_matrix_access m r' c' builder locals_map mat_dim_map), (mat_dim_map,
img_dim_map))
| ImageLit (m1, m2, m3)  -> (* m1, m2, and m3 are all IDs of matrices with dimensions
(4032,3024) *)
                            (* Allocate space for the struct *)
                            let img = L.build_alloca (ltype_of_typ (Image)) "img"
builder
                            (* Get pointers to every element in struct *)
                            in let m1_pointer = L.build_struct_gep img 0 "m1_pointer"
builder
                            (* m1 is element ptr to img at idx 0*)
                            and m2_pointer = L.build_struct_gep img 1 "m2_pointer"
builder
                            and m3_pointer = L.build_struct_gep img 2 "m3_pointer"
builder
                            (* Get the pointer to the DimMat m1, m2, and m3 *)
                            in let m1' = StringMap.find m1 locals_map
                            (* m1' is the pointer to the LLValue of the mat *)
                            and m2' = StringMap.find m2 locals_map
                            and m3' = StringMap.find m3 locals_map
                            (* Get the pointer to the DimMat m1, m2, and m3 *)
                            in let m1_val = L.build_load m1' "m1_val" builder
                            and m2_val = L.build_load m2' "m2_val" builder
                            and m3_val = L.build_load m3' "m3_val" builder
                            (* Store appropriate values into the pointers *)
                            in ignore(L.build_store m1_val m1_pointer builder);
                            ignore(L.build_store m2_val m2_pointer builder) ;
                            ignore(L.build_store m3_val m3_pointer builder) ;
                            ((L.build_load (L.build_gep img [| L.const_int i32_t 0|]
                                "img" builder ) "img" builder), (mat_dim_map,
img_dim_map))
| ImageRedAccess img_id  -> let img_val = StringMap.find img_id locals_map
                            in let pointer_to_red = L.build_struct_gep img_val 0
"i_red" builder
                            in ((L.build_load pointer_to_red "actual_red" builder),
(mat_dim_map, img_dim_map))
| ImageGreenAccess img_id-> let img_val = StringMap.find img_id locals_map
                            in let pointer_to_green = L.build_struct_gep img_val 1
```

```
"i_green" builder
                            in ((L.build_load pointer_to_green "actual_green"
builder), (mat_dim_map, img_dim_map))
| ImageBlueAccess img_id -> let img_val = StringMap.find img_id locals_map
                            in let pointer_to_blue = L.build_struct_gep img_val 2
"i_blue" builder
                            in ((L.build_load pointer_to_blue "actual_blue" builder),
(mat_dim_map, img_dim_map))
| MatrixRowSize m        -> let size = StringMap.find m mat_dim_map
                            in (L.const_int i32_t (fst (size)), (mat_dim_map,
img_dim_map))
| MatrixColSize m        -> let size = StringMap.find m mat_dim_map
                            in (L.const_int i32_t (snd (size)), (mat_dim_map,
img_dim_map))
| Cast (_, _)            -> (L.const_int i32_t 10, (mat_dim_map, img_dim_map)) (*
TODO jk  TO REMOVE *)
| Call ("save", [e])     ->
      let img_struct_alloc = L.build_alloca image_t "tmp_img_alloc" builder
      in let img_struct_val = fst (expr locals_map mat_dim_map img_dim_map builder e)
      in ignore(L.build_store img_struct_val img_struct_alloc builder);

      let pointer_to_red = L.build_struct_gep img_struct_alloc 0 "i_red" builder
      in let _  = L.build_load pointer_to_red "actual_red" builder
      in let pointer_to_blue = L.build_struct_gep img_struct_alloc 2 "i_blue" builder
      in let _ = L.build_load pointer_to_blue "actual_blue" builder
      in let pointer_to_green = L.build_struct_gep img_struct_alloc 1 "i_green"
builder
      in let _ = L.build_load pointer_to_green "actual_green" builder

      in let red_mat_ptr = L.build_gep pointer_to_red [| L.const_int i32_t 0;
L.const_int i32_t 0 |] "ptr_red" builder
      in let blue_mat_ptr = L.build_gep pointer_to_blue [| L.const_int i32_t 0;
L.const_int i32_t 0 |] "ptr_blue" builder
      in let green_mat_ptr = L.build_gep pointer_to_green [| L.const_int i32_t 0;
L.const_int i32_t 0 |] "ptr_green" builder

      in let ptr_typ = L.pointer_type (array_t double_t image_row_size)
      in let save_cpp_t = L.function_type void_t [| ptr_typ; ptr_typ; ptr_typ |]
      in let save_cpp_func = L.declare_function "save_cpp" save_cpp_t the_module

      in ignore(L.build_call save_cpp_func [| red_mat_ptr; green_mat_ptr;
blue_mat_ptr |] "" builder);
      (L.const_int i32_t 0, (mat_dim_map, img_dim_map)) (* return void *)
| Call ("print", [e])    ->
      let e' = fst (expr locals_map mat_dim_map img_dim_map builder e)
      in let ltyp = L.string_of_lltype (L.type_of e')
      in if check_if_matrix ltyp then (match e with
          Id m ->
            let (m_row, m_col) = StringMap.find m mat_dim_map
            in  print_by_type ltyp e' builder mat_dim_map img_dim_map m_row m_col
        | _ -> raise (Failure ("Cannot print anonymous matrix")))
      else print_by_type ltyp e' builder mat_dim_map img_dim_map (-1) (-1)
| Call (fname, act) ->
      (try let (fdef, fdecl) = StringMap.find fname function_decls
        in let actuals = List.rev (List.map fst (List.map
          (expr locals_map mat_dim_map img_dim_map builder) (List.rev act))  )
        in let result = (match fdecl.typ with
            Void -> ""
          | _ -> fname ^ "_result")
          in (try ((L.build_call fdef (Array.of_list actuals) result builder),
(mat_dim_map, img_dim_map))
```

```
            with _ -> raise (Failure ("Call function failed on " ^ fname)))
          with _ -> raise(Failure ("cannot find function " ^ fname)))
    in


(* SECTION 8: STMT *)
(* Returns (builder, locals_map) tuple *)
  let rec stmt (the_function, fdecl) (builder, (locals_map, (mat_dim_map, img_dim_map)))
    = function
     Block sl -> List.fold_left (stmt (the_function, fdecl)) (builder, (locals_map,
    (mat_dim_map, img_dim_map))) sl
    | Expr e -> ignore(fst (expr locals_map mat_dim_map img_dim_map builder e));
    (builder, (locals_map, (mat_dim_map, img_dim_map)))
    | Local (typ, name, e) ->
        (* If local assigning a matrix, we need to calculate the matrix dimensions to
    allocate memory. *)
        (match typ with
            Matrix  ->
              (match e with
                  Noassign (_) ->
                    (* Allocate a 0x0 dimension matrix for a matrix declaration (no size)
    e.g. matrix m; *)
                      let local_var = L.build_alloca (ltype_of_typ (DimMatrix(0,0))) name
    builder
                      in let new_locals_map = StringMap.add name local_var locals_map
                      in (builder, (new_locals_map, (mat_dim_map, img_dim_map)))
                | MatLit m ->
                      let (row, col) = get_matrix_dim m
                      in let new_mat_dim_map = StringMap.add name (row, col) mat_dim_map
                      in let mat_alloc = L.build_alloca (ltype_of_typ (DimMatrix(row,
    col))) name builder
                      in let mat_val = (fst (expr locals_map new_mat_dim_map img_dim_map
    builder e))
                      in let new_locals_map = StringMap.add name mat_alloc locals_map
                      in ignore(L.build_store mat_val mat_alloc builder);
                      (builder, (new_locals_map, (new_mat_dim_map, img_dim_map)))
                | Id idName ->
                     (try let _ = StringMap.find idName locals_map
                        and (row, col) = StringMap.find idName mat_dim_map
                        in let mat_alloc = L.build_alloca (ltype_of_typ (DimMatrix(row,
    col))) name builder
                        in let new_locals_map = StringMap.add name mat_alloc locals_map
                        and new_mat_dim_map = StringMap.add name (row, col) mat_dim_map
                        in let mat_val = (fst (expr new_locals_map new_mat_dim_map
    img_dim_map builder e))
                        in ignore (L.build_store mat_val mat_alloc builder);
                        (builder, (new_locals_map, (new_mat_dim_map, img_dim_map)))
                      with Not_found -> raise (Failure ("Variable not found 5: " ^
    idName)))
                | Binop(m1, _, m2) as e ->
                      let assign_mat_to_binop =
                      let (new_matrix, (new_mat_dim_map, _)) = expr locals_map
    mat_dim_map img_dim_map builder e
                    in (try let (row, col) = StringMap.find "binop_result"
    new_mat_dim_map
                        in let mat_alloc = L.build_alloca (L.type_of new_matrix) name
    builder
                        in let new_locals_map = StringMap.add name mat_alloc locals_map
                        and new_mat_dim_map = StringMap.add name (row, col) mat_dim_map
                        in let mat_val = fst (expr new_locals_map new_mat_dim_map
    img_dim_map builder e)
```

```
                    in ignore (L.build_store mat_val mat_alloc builder);
                    (builder, (new_locals_map, (new_mat_dim_map, img_dim_map)))
                    with Not_found -> raise (Failure ("Variable not found 6: " ^
name)))
                  in (match (m1, m2) with
                      (Id _, Id _) -> assign_mat_to_binop
                    | (Id _, DblLit _) -> assign_mat_to_binop
                    | (DblLit _, Id _) -> assign_mat_to_binop
                    | _ -> raise (Failure "Can't perform binop on unnamed matrices")
                )
              | ImageRedAccess _ | ImageGreenAccess _ | ImageBlueAccess _ ->
                    let (row, col) = (image_row_size,image_col_size)
                    in let new_mat_dim_map = StringMap.add name (row, col) mat_dim_map
                    in let mat_alloc = L.build_alloca (ltype_of_typ (DimMatrix(row,
col))) name builder
                    in let mat_val = (fst (expr locals_map new_mat_dim_map img_dim_map
builder e))
                    in let new_locals_map = StringMap.add name mat_alloc locals_map
                    in ignore(L.build_store mat_val mat_alloc builder);
                    (builder, (new_locals_map, (new_mat_dim_map, img_dim_map)))
              | Call (fname, act) ->
                   (match fname with
                     "dim" ->
                         let actuals = List.rev (List.map fst (List.map (expr
locals_map mat_dim_map
                             img_dim_map builder) (List.rev act)))
                         in let dim_cpp_t = L.function_type (L.pointer_type double_t)
[| str_t |]
                         in let dim_cpp_func = L.declare_function "dim_cpp" dim_cpp_t
the_module
                         in let path = List.hd actuals
                         in let dim_return = L.build_call dim_cpp_func [| path |]
"dim_ret" builder
                         in (* double* *) let row = L.build_load (L.build_gep
dim_return
                             [| L.const_int i32_t 0 |] "row_ptr" builder ) "row_val"
builder
                         in let col = L.build_load (L.build_gep dim_return [|
L.const_int i32_t 1 |]
                             "col_ptr" builder ) "col_val" builder
                         in let mat_alloc = L.build_alloca (ltype_of_typ (DimMatrix(1,
2))) "tmp" builder
                         in let mat_row = L.build_gep mat_alloc [| L.const_int i32_t
0; L.const_int i32_t 0;
                             L.const_int i32_t 0 |] "" builder
                         in let mat_col = L.build_gep mat_alloc [| L.const_int i32_t
0; L.const_int i32_t 0;
                             L.const_int i32_t 1 |] "" builder
                         in let new_locals_map = StringMap.add name mat_alloc
locals_map
                         in let new_mat_dim_map = StringMap.add name (1, 2) mat_dim_map
                         in ignore(L.build_store row mat_row builder);
                         ignore(L.build_store col mat_col builder);
                         (builder, (new_locals_map, (new_mat_dim_map, img_dim_map)))
                    | _ ->
                         (try let (fdef, fdecl) = StringMap.find fname function_decls
                             in let actuals = List.rev (List.map fst (List.map (expr
locals_map mat_dim_map
                                 img_dim_map builder) (List.rev act))  )
                             in let result = (match fdecl.typ with
                                   Void -> ""
```

```
                                | _ -> fname ^ "_result")
                            in (try let mat_val = ((L.build_call fdef (Array.of_list
actuals) result builder))
                              in let new_mat_dim_map = StringMap.add name (1, 2)
mat_dim_map
                              in let mat_alloc = L.build_alloca (ltype_of_typ
(DimMatrix(1, 2))) name builder
                              in let new_locals_map = StringMap.add name mat_alloc
locals_map
                              in ignore (L.build_store mat_val mat_alloc builder);
                              (builder, (new_locals_map, (new_mat_dim_map,
img_dim_map)))
                            with _ -> raise (Failure ("Image call function failed on
" ^ fname)))
                        with _ -> raise(Failure ("Image Cannot find function " ^
fname))))
          | _ -> raise (Failure ("Can't assign non matrix to a matrix (either
noassign, matlit, id)")))
      | Image  ->
          (match e with
              Noassign (_) ->
                  let local_var = L.build_alloca (ltype_of_typ (Image)) name builder
                  in let new_locals_map = StringMap.add name local_var locals_map
                  in (builder, (new_locals_map, (mat_dim_map, img_dim_map)))
            | ImageLit (_, _, _) ->
                  let (row, col) = (image_row_size,image_col_size)
                  in let new_img_dim_map = StringMap.add name (row, col) img_dim_map
                  in let img_alloc = L.build_alloca (ltype_of_typ (Image)) name
builder
                  and img_val = (fst (expr locals_map mat_dim_map new_img_dim_map
builder e))
                  in let new_locals_map = StringMap.add name img_alloc locals_map
                  in ignore(L.build_store img_val img_alloc builder);
                  (builder, (new_locals_map, (mat_dim_map, new_img_dim_map)))
            | Id idName ->
                  (try let _ = StringMap.find idName locals_map
                    and (row, col) = (image_row_size,image_col_size)
                    in let img_alloc = L.build_alloca (ltype_of_typ (Image)) name
builder
                    in let new_locals_map = StringMap.add name img_alloc locals_map
                    and new_img_dim_map = StringMap.add name (row, col) img_dim_map
                    in let image_val = (fst (expr new_locals_map mat_dim_map
new_img_dim_map builder e))
                    in ignore (L.build_store image_val img_alloc builder);
                    (builder, (new_locals_map, (mat_dim_map, new_img_dim_map)))
                  with Not_found -> raise (Failure ("Variable not found 1: " ^
idName)))
            | Call (fname, act) ->
                  (match fname with
                    | "edgedetect" ->
                        let actuals = List.rev (List.map fst (List.map (expr
locals_map mat_dim_map
                            img_dim_map builder) (List.rev act))  )
                        in let edgedetect_cpp_t = L.function_type (L.pointer_type
double_t) [| str_t |]
                        in let edgedetect_cpp_func = L.declare_function
"edgedetect_cpp" edgedetect_cpp_t
                            the_module
                        in let path = List.hd actuals
                        in let edgedetect_return = L.build_call edgedetect_cpp_func
[| path |]
```

```
                            "edgedetect_ret" builder
                        in load_image name edgedetect_return mat_dim_map img_dim_map
locals_map builder
                    | "grayscale" ->
                        let actuals = List.rev (List.map fst (List.map (expr
locals_map mat_dim_map
                            img_dim_map builder) (List.rev act))  )
                        in let grayscale_cpp_t = L.function_type (L.pointer_type
double_t) [| str_t |]
                        in let grayscale_cpp_func = L.declare_function
"grayscale_cpp" grayscale_cpp_t the_module
                        in let path = List.hd actuals
                        in let grayscale_return = L.build_call grayscale_cpp_func [|
path |]
                            "grayscale_ret" builder
                        in load_image name grayscale_return mat_dim_map img_dim_map
locals_map builder
                    | "brighten" ->
                        let actuals = List.rev (List.map fst (List.map (expr
locals_map mat_dim_map
                            img_dim_map builder) (List.rev act))  )
                        in let brighten_cpp_t = L.function_type (L.pointer_type
double_t) [| str_t |]
                        in let brighten_cpp_func = L.declare_function "brighten_cpp"
brighten_cpp_t the_module
                        in let path = List.hd actuals
                        in let brighten_return = L.build_call brighten_cpp_func [|
path |] "brighten_ret" builder
                        in load_image name brighten_return mat_dim_map img_dim_map
locals_map builder
                    | "blur" ->
                        let actuals = List.rev (List.map fst (List.map (expr
locals_map mat_dim_map
                            img_dim_map builder) (List.rev act))  )
                        in let blur_cpp_t = L.function_type (L.pointer_type double_t)
[| str_t |]
                        in let blur_cpp_func = L.declare_function "blur_cpp"
blur_cpp_t the_module
                        in let path = List.hd actuals
                        in let blur_return = L.build_call blur_cpp_func [| path |]
"blur_ret" builder
                        in load_image name blur_return mat_dim_map img_dim_map
locals_map builder
                    | "load" ->
                        let actuals = List.rev (List.map fst (List.map (expr
locals_map mat_dim_map
                            img_dim_map builder) (List.rev act))  )
                        in let load_cpp_t = L.function_type (L.pointer_type double_t)
[| str_t |]
                        in let load_cpp_func = L.declare_function "load_cpp"
load_cpp_t the_module
                        in let path = List.hd actuals
                        in let load_return = L.build_call load_cpp_func [| path |]
"load_ret" builder
                        in (* double* *) load_image name load_return mat_dim_map
img_dim_map locals_map builder
                    | _ -> (try let (fdef, fdecl) = StringMap.find fname
function_decls in
                        let actuals = List.rev (List.map fst (List.map (expr
locals_map mat_dim_map
                            img_dim_map builder) (List.rev act))  )
```

```
                        in let result =
                            (match fdecl.typ with
                                Void -> ""
                              | _ -> fname ^ "_result")
                        in (try let image_val = ((L.build_call fdef (Array.of_list
actuals) result
                            builder)) in let img_alloc = L.build_alloca (ltype_of_typ
(Image)) name builder
                        in let new_locals_map = StringMap.add name img_alloc
locals_map
                        in ignore (L.build_store image_val img_alloc builder);
                        (builder, (new_locals_map, (mat_dim_map, img_dim_map)))
                        with _ -> raise (Failure ("Image call function failed on " ^
fname)))
                  with _ -> raise(Failure ("Image Cannot find function " ^ fname))))
                | _ -> raise (Failure ("Can't assign non image to a image (either
noassign, imagelit, id)"))
             )
        (* Non Matrix/Image*)
      | _       ->
          let local_var = L.build_alloca (ltype_of_typ typ) name builder
          in let new_locals_map = StringMap.add name local_var locals_map
          in ignore (L.build_store (fst (expr new_locals_map mat_dim_map img_dim_map
builder e)) local_var builder);
          (builder, (new_locals_map, (mat_dim_map, img_dim_map)))
    )
| Return e -> ignore(match fdecl.typ with
                  (* Special "return nothing" instr *)
                  Void -> L.build_ret_void builder
                  (* Build return statement *)
                | _ -> L.build_ret (fst (expr locals_map mat_dim_map img_dim_map
builder e)) builder );
              (builder, (locals_map, (mat_dim_map, img_dim_map)))
| If (predicate, then_stmt, else_stmt) ->
       let bool_val = fst (expr locals_map mat_dim_map img_dim_map builder predicate)
in
       let merge_bb = L.append_block context "merge" the_function in
            let build_br_merge = L.build_br merge_bb in (* partial function *)

       let then_bb = L.append_block context "then" the_function in
       add_terminal (stmt (the_function, fdecl) ((L.builder_at_end context then_bb),
(locals_map,
            (mat_dim_map, img_dim_map))) then_stmt)
         build_br_merge;

       let else_bb = L.append_block context "else" the_function in
       add_terminal (stmt (the_function, fdecl) ((L.builder_at_end context else_bb),
(locals_map,
            (mat_dim_map, img_dim_map))) else_stmt)
         build_br_merge;

       ignore(L.build_cond_br bool_val then_bb else_bb builder);
       (L.builder_at_end context merge_bb, (locals_map, (mat_dim_map, img_dim_map)))
| While (predicate, body) ->
     let pred_bb = L.append_block context "while" the_function in
     ignore(L.build_br pred_bb builder);

     let body_bb = L.append_block context "while_body" the_function in
     add_terminal (stmt (the_function, fdecl) ((L.builder_at_end context body_bb),
(locals_map,
            (mat_dim_map, img_dim_map))) body)
```

```
            (L.build_br pred_bb);

          let pred_builder = L.builder_at_end context pred_bb in
          let bool_val = fst (expr locals_map mat_dim_map img_dim_map pred_builder
    predicate) in

          let merge_bb = L.append_block context "merge" the_function in
          ignore(L.build_cond_br bool_val body_bb merge_bb pred_builder);
          (L.builder_at_end context merge_bb, (locals_map, (mat_dim_map, img_dim_map)))
    | For (e1, e2, e3, body) ->
          stmt (the_function, fdecl) (builder, (locals_map, (mat_dim_map, img_dim_map)))
    ( Block [Expr e1 ;
              While (e2, Block [body ; Expr e3]) ] )

  in


(* SECTION 9: Fill in the body of the given function *)
  let build_function_body fdecl =
    let (the_function, _) = StringMap.find fdecl.fname function_decls
    in let builder_for_stmt = L.builder_at_end context (L.entry_block the_function)


    (* Set formal arguments *)
    in let add_formal m (t, n) p =
      L.set_value_name n p;
    let local = L.build_alloca (ltype_of_typ t) n builder_for_stmt
        in ignore (L.build_store p local builder_for_stmt);
    StringMap.add n local m

    in let formals = List.fold_left2 add_formal StringMap.empty fdecl.formals (* list of
    tuples (typ, name) *)
        (Array.to_list (L.params the_function)) (* list of the actual parameter values *)

    (* Build the code for each statement in the function. The StringMap.empty as the
    second value in the second tuple is a StringMap to keep track of matrix sizes *)
    in let builder_and_3maps_tuple = stmt (the_function, fdecl) (builder_for_stmt,
    (formals, (StringMap.empty,
        StringMap.empty))) (Block fdecl.body)

    (* Add a return if the last block falls off the end *)
    in add_terminal builder_and_3maps_tuple (match fdecl.typ with
        Void -> L.build_ret_void
      | Double -> L.build_ret (L.const_float double_t 0.0)
      | t -> L.build_ret (L.const_int (ltype_of_typ t) 0))
  in


List.iter build_function_body functions; the_module
```

10.6 Makefile

```
vscode.native :
    opam config exec -- \
    ocamlbuild -use-ocamlfind vscode.native

TARFILES = Makefile scanner.mll parser.mly ast.ml vscode.ml semant.ml codegen.ml _tags
```

```
OBJS = ast.cmx codegen.cmx parser.cmx scanner.cmx semant.cmx vscode.cmx

LINKS = ast.cmo semant.cmo

vscode : $(OBJS) $(LINKS)
    ocamlc $(LINKS) -o vscode $(OBJS)

scanner.ml : scanner.mll
    ocamllex scanner.mll

parser.ml parser.mli : parser.mly
    ocamlyacc -v parser.mly

%.cmo : %.ml
    ocamlc -c $<

%.cmi : %.mli
    ocamlc -c $<

vscode.tar.gz : $(TARFILES)
    cd .. && tar zcf vscode/vscode.tar.gz $(TARFILES:%=vscode/%)

.PHONY : clean
clean :
    rm -f vscode parser.ml parser.mli scanner.ml *.native *.cmo *.cmi parser.output
    test-.* fail-.*

.Phony : menhir
menhir:
    menhir --interpret --interpret-show-cst parser.mly

# Generated by ocamldep *.ml *.mli
ast.cmo:
ast.cmx:
vscode.cmo: scanner.cmo parser.cmi ast.cmo semant.cmo codegen.cmo
vscode.cmx: scanner.cmx parser.cmx ast.cmx semant.cmx codegen.cmx
codegen.cmo : ast.cmo
codegen.cmx : ast.cmx
semant.cmo: ast.cmo
semant.cmx: ast.cmx
parser.cmi: ast.cmo
parser.cmo: ast.cmo parser.cmi
parser.cmx: ast.cmx parser.cmi
scanner.cmo: parser.cmi
scanner.cmx: parser.cmx
```

10.7 utils.cpp

```
/*
File: UTILS.CPP
Description: Defines all C++ functions used
*/

#include <opencv2/core.hpp>
#include <opencv2/imgcodecs.hpp>
#include <opencv2/highgui.hpp>
#include <opencv2/opencv.hpp>
#include "opencv2/objdetect/objdetect.hpp"
```

```
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"

#include <stdio.h>
#include <iostream>
#include <string>

using namespace cv;
using namespace std;

int width = 5;
int height = 5;


extern "C" int row_size_cpp (char imageName[]) {
    Mat img = imread(imageName,CV_LOAD_IMAGE_COLOR);
    int rows = img.rows;
    return rows;
}


extern "C" int* dbl_arr_cpp () {
    int* foo = new int[2];
    foo[0] = 1;
    foo[1] = 2;
    return foo;
}


extern "C" double* dim_cpp (char imageName[]) {
    Mat img = imread(imageName,CV_LOAD_IMAGE_COLOR);
    double* output = new double[2];
    output[0] = (double) img.rows;
    output[1] = (double) img.cols;
    return output;
}


extern "C" double* load_cpp(char imageName[]) {
    Mat img = imread(imageName,CV_LOAD_IMAGE_COLOR);

    unsigned char* input = (unsigned char*)(img.data);
    double* output = new double[3*height*width];

    double r,g,b;
    int rows = img.rows;
    int cols = img.cols;
    int k = 0;
    for(int i = 0; i < width; i++){
        for(int j = 0; j < height; j++){
            if ( i >= rows || j >= cols ) {
              b = 0;
                output[k++] = b;
                g = 0;
                output[k++] = g;
                r = 0;
                output[k++] = r;
               } else {
                b = input[img.step * i + j*img.channels()] ;
                output[k++]=b;
                g = input[img.step * i + j*img.channels() + 1];
```

```
                output[k++]=g;
                r = input[img.step * i + j*img.channels() + 2];
                output[k++]=r;
               }
        }
    }
    return output;
}


extern "C" double* blur_cpp(char imageName[]) {
    Mat src = imread(imageName,CV_LOAD_IMAGE_COLOR);
    Mat img;

    blur(src, img, Size(10,10));

    unsigned char* input = (unsigned char*)(img.data);
    double* output = new double[3*height*width];

    double r,g,b;
    int rows = img.rows;
    int cols = img.cols;
    int k = 0;
    for(int i = 0; i < width; i++){
        for(int j = 0; j < height; j++){
            if ( i >= rows || j >= cols ) {
                b = 0;
                output[k++] = b;
                g = 0;
                output[k++] = g;
                r = 0;
                output[k++] = r;
            } else {
                b = input[img.step * i + j*img.channels()] ;
                output[k++]=b;
                g = input[img.step * i + j*img.channels() + 1];
                output[k++]=g;
                r = input[img.step * i + j*img.channels() + 2];
                output[k++]=r;
            }
        }
    }
    return output;
}


extern "C" double* brighten_cpp(char imageName[]) {
    Mat src = imread(imageName,CV_LOAD_IMAGE_COLOR);
    Mat img;

    bilateralFilter(src, img, 15, 80, 80, BORDER_DEFAULT);

    unsigned char* input = (unsigned char*)(img.data);
    double* output = new double[3*height*width];

    double r,g,b;
    int rows = img.rows;
    int cols = img.cols;
    int k = 0;
    for(int i = 0; i < width; i++){
        for(int j = 0; j < height; j++){
```

```
            if ( i >= rows || j >= cols ) {
                b = 0;
                output[k++] = b;
                g = 0;
                output[k++] = g;
                r = 0;
                output[k++] = r;
            } else {
                b = input[img.step * i + j*img.channels()] ;
                output[k++]=b;
                g = input[img.step * i + j*img.channels() + 1];
                output[k++]=g;
                r = input[img.step * i + j*img.channels() + 2];
                output[k++]=r;
            }
        }
    }
    return output;
}


extern "C" double* grayscale_cpp(char imageName[]) {
    Mat src = imread(imageName,CV_LOAD_IMAGE_COLOR);
    Mat img;

    cvtColor(src, img, COLOR_RGB2GRAY);

    unsigned char* input = (unsigned char*)(img.data);
    double* output = new double[3*height*width];

    double r,g,b;
    int rows = img.rows;
    int cols = img.cols;
    int k = 0;
    for(int i = 0; i < width; i++){
        for(int j = 0; j < height; j++){
            if ( i >= rows || j >= cols ) {
                b = 0;
                output[k++] = b;
                g = 0;
                output[k++] = g;
                r = 0;
                output[k++] = r;
            } else {
                b = input[img.step * i + j*img.channels()] ;
                output[k++]=b;
                g = input[img.step * i + j*img.channels() + 1];
                output[k++]=g;
                r = input[img.step * i + j*img.channels() + 2];
                output[k++]=r;
            }
        }
    }
    return output;
}


extern "C" double* edgedetect_cpp(char imageName[]) {
    Mat src = imread(imageName,CV_LOAD_IMAGE_COLOR);
    Mat mid;
    Mat img;
```

```
    cvtColor(src, mid, COLOR_RGB2GRAY);
    Canny(mid, img, 60, 60*3);

    unsigned char* input = (unsigned char*)(img.data);
    double* output = new double[3*height*width];

    double r,g,b;
    int rows = img.rows;
    int cols = img.cols;
    int k = 0;
    for(int i = 0; i < width; i++){
        for(int j = 0; j < height; j++){
            if ( i >= rows || j >= cols ) {
                b = 0;
                output[k++] = b;
                g = 0;
                output[k++] = g;
                r = 0;
                output[k++] = r;
            }
            else {
                b = input[img.step * i + j*img.channels()] ;
                output[k++]=b;
                g = input[img.step * i + j*img.channels() + 1];
                output[k++]=g;
                r = input[img.step * i + j*img.channels() + 2];
                output[k++]=r;
            }
        }
    }
    return output;
}


extern "C" void save_cpp(double r[height][width], double g[height][width], double
    b[height][width]) {
    int h = height;
    int w = width;
    double* data = new double[3*h*w];

    for (int i = 0; i < w; i++) {
        for (int j = 0; j < h; j++) {
            data[3*(h*i+j)] = b[i][j];
            data[3*(h*i+j)+1] = g[i][j];
            data[3*(h*i+j)+2] = r[i][j];
        }
    }

    Mat image = cv::Mat(h, w, CV_64FC3, data);
        imwrite("image_out.jpg",image);
        return ;
}
```

10.8 link.sh

```
#!/bin/bash
```

```
# File: LINK.SH
# Compiles the .vsc file into LLVM ir,
# then static compiles with llc-5.0,
# then links in utils.cpp to finally generate executable

for var in "$@"
do
    basename=‘echo $var | sed ’s/.*\\///
                              s/.vsc//’‘
    rm ${basename}.ir;
    rm ${basename}.exe;
    ./vscode.native $var.vsc >> ${basename}.ir;
    llc-5.0 ${basename}.ir;
    clang++-4.0 ‘pkg-config --cflags opencv‘ ‘pkg-config --libs opencv‘ ${basename}.ir.s
    utils.cpp -o ${basename}.exe
done
```