

GWiz Final Report

Katherine Duff kpd2128, Ashley Kim atk2141,
Elisa Luo eyl2130, Rebecca Yao rby2107

April 26, 2021

Contents

| | | |
|----------|--|----------|
| 1 | Introduction | 4 |
| 2 | Tutorial | 4 |
| 2.1 | Environment Setup | 4 |
| 2.2 | Language Tutorial | 4 |
| 2.3 | Compiling and Executing a GWiz program | 5 |
| 3 | Language Reference Manual | 5 |
| 3.1 | Lexical Conventions | 5 |
| 3.1.1 | Tokens | 5 |
| 3.1.2 | Comments | 5 |
| 3.1.3 | Separators | 5 |
| 3.1.4 | Operators | 6 |
| 3.1.5 | Identifiers | 6 |
| 3.1.6 | Keywords | 6 |
| 3.1.7 | Literals | 6 |
| 3.2 | Data Types | 7 |
| 3.2.1 | Primitives | 7 |
| 3.2.2 | Mutability | 8 |
| 3.3 | Built In Types | 8 |
| 3.3.1 | Trackpoint | 8 |
| 3.3.2 | Linked List | 8 |
| 3.4 | Scoping Rules | 10 |
| 3.5 | Memory | 10 |
| 3.6 | Expressions and Operators | 10 |
| 3.6.1 | Unary Operators | 10 |
| 3.6.2 | Binary Operators | 10 |
| 3.6.3 | Operator Precedence | 11 |
| 3.7 | Statements | 11 |
| 3.7.1 | If else Statement | 12 |
| 3.7.2 | For Statement | 12 |

| | | |
|-----------|--|-----------|
| 3.7.3 | Return Statement | 12 |
| 3.8 | While Statement | 13 |
| 3.9 | Functions | 13 |
| 3.9.1 | Function Declarations | 13 |
| 3.9.2 | Function Calls | 13 |
| 3.9.3 | Variable assignment from functions | 13 |
| 3.9.4 | General Functions | 13 |
| 3.9.5 | Trackpoint Functions | 14 |
| 3.9.6 | GPX File Functions | 14 |
| 4 | Project Plan | 15 |
| 4.1 | Planning, Specification, Development and Testing | 15 |
| 4.2 | Style Guide | 16 |
| 4.3 | Project Timeline | 17 |
| 4.4 | Roles and Responsibilities | 17 |
| 4.5 | Software Development Environment | 17 |
| 5 | Language Evolution | 17 |
| 6 | Architectural Design | 18 |
| 6.1 | The Compiler | 18 |
| 6.2 | The Lexer | 18 |
| 6.3 | The Parser | 19 |
| 6.4 | Semantic Checking | 19 |
| 6.5 | Code Generation | 19 |
| 7 | C Libraries | 19 |
| 7.1 | parse_gpx.c | 19 |
| 7.2 | List.c | 19 |
| 7.3 | struct.c | 19 |
| 8 | Test Plan | 20 |
| 8.1 | Unit Testing | 20 |
| 8.2 | Integration Tests, Test Automation and Scripts | 20 |
| 8.3 | Example Test Programs | 20 |
| 9 | Lessons Learned | 26 |
| 10 | Appendix | 28 |
| 10.1 | Project Log | 28 |
| 10.2 | gwiz.ml | 61 |
| 10.3 | scanner.mll | 62 |
| 10.4 | gwizparse.mly | 64 |
| 10.5 | ast.ml | 67 |
| 10.6 | sast.ml | 70 |
| 10.7 | semant.ml | 73 |
| 10.8 | codegen.ml | 79 |

| | | |
|-------|----------------------------|-----|
| 10.9 | struct.c | 91 |
| 10.10 | struct.h | 91 |
| 10.11 | list.c | 92 |
| 10.12 | parse_gpx.c | 93 |
| 10.13 | Makefile | 100 |
| 10.14 | testall.sh | 102 |
| 10.15 | Test Suite | 106 |
| | 10.15.1 Fail Test Files | 106 |
| | 10.15.2 Success Test Files | 114 |

1 Introduction

GWiz (GPX Wizard) is an imperative, statically scoped and strongly typed programming language and allows for the analysis of GPX files. GPX files are generated by using a watch, phone, or other device to track a run, walk, swim, hike, or bike ride.

2 Tutorial

2.1 Environment Setup

To run a program in GWiz, you'll need the `libxml2-dev` package. Installation may vary by system. In a Docker container, you can run:

```
1 apt-get update
2 apt-get install -y libxml2-dev
```

2.2 Language Tutorial

Having set up the environment, let's review how to program in GWiz:

- The file extension for GWiz files is “.gw”
- A program can have global variables at the beginning, followed by functions. It must have a main function, which will be the starting point of statement execution.
- Variables are declared as `<type> varname;`
- Functions are declared as `<type> functionname(opt. <type> arg1,){vars statements` and contain local variables at the beginning followed by statements. They end with a `return` statement.
- Variables must be initialized. This must be a separate statement and not part of the variable declaration.
- Statement blocks are surrounded by `{}`
- Statements end in `;` unless they have a statement block such as a for loop.
- It's best to use `{` to avoid confusion in loop or conditional statements.
- It's best to use `()` to disambiguate operations.
- Use `/* */` for comments.

2.3 Compiling and Executing a GWiz program

Now that we've written a simple program in GWiz, let's run it through the GWiz compiler. First, build the source compiler. Navigate to the `gwiz` folder and run `make all`. Then, to execute a GWiz program, run:

```
1 $ ./gwiz <program.gw>
```

3 Language Reference Manual

3.1 Lexical Conventions

3.1.1 Tokens

After lexical translation, characters are reduced to a sequence of input elements, those being white space, comments, and tokens. Tokens are divided into identifiers, separators, operators, GWiz keywords, and literals.

3.1.2 Comments

GWiz has both single-line and multi-line comments. They are both denoted using the `/*` and `*/` symbols. Comments start with the `/*` characters and end with the `*/` characters, ignoring all characters between the start and ending characters.

```
1 /* This is a single line comment. Text in here is ignored */
```

```
1 /* This is a multi line comment.  
2 Text in here  
3 is ignored */
```

Comments cannot be nested.

3.1.3 Separators

The following 9 tokens are the separators in GWiz.

```
1 ( ) { } [ ] ; ,
```

- `()` is used for precedence in expressions and denoting arguments in function calls
- `{ }` is used for scoping
- `[]` is used for array indices
- `;` denotes the end of a line and separates statements
- `,` is used to separate variables or parameters

3.1.4 Operators

The following 14 tokens are the operators in GWiz. These are the arithmetic, assignment, comparison, and logical operators in GWiz.

```
1 + - * /
2 =
3 == < > != >= <=
4 ! && ||
```

3.1.5 Identifiers

An identifier in GWiz is an unlimited length sequence of ASCII letters [A-Z, a-z] and decimal digits [0-9]. It must begin with a letter, cannot be a GWiz keyword, or contain a character other than a letter or digit (e.g. student_name is not a valid identifier)

Convention for identifiers is camel case, where the first letter is lowercase and every subsequent first character of a word is uppercase. (e.g. studentName)

The regex for identifiers following GWiz convention is `[a-z][0-9a-zA-Z]*`

3.1.6 Keywords

The following identifiers are reserved for use as keywords and may not be used as identifiers.

| Use | Reserved words | | | |
|------------------------|--------------------------------------|-----------------------------|----------|--|
| booleans | true | false | | |
| control flow | if | else | for | while |
| functions | void | return | | |
| data types | int | double | string | bool |
| data structures | Trackpoint | file | | |
| built ins | prints printi printd printb | getLat getLon getTime | parseGPX | totTime totDist avgSpeed stat |

Note, "true"/"false" are not specifically keywords in GWiz, but instead boolean literals. Void has no associated value and can only be used as a return type for a function that returns nothing.

3.1.7 Literals

Integer Literals

A sequence of one or more numerical digits representing an integer. The matching regex is `[-]?[0-9]+`

Examples of integer literals:

```
1 0
2 5
3 163
4 -42
```

Double Literals

A sequence of zero or more numerical digits followed by a '.', followed by one or more numerical digits.

The matching regex is `[-]?[0-9]*'.'[0-9]+`

Examples of double literals:

```
1 1.4
2 .5
3 0.5829
4 -1.5
```

Boolean Literals

Boolean types represent true and false. They are represented as `true` and `false` in GWiz.

Boolean literals:

```
1 true
2 false
```

String Literals

A sequence of characters enclosed in double quotes. Escape sequences may be present between the double quotes.

Examples of string literals:

```
1 string str = "hello world!";
2 string c = "c";
3 string empty = "";
```

3.2 Data Types

There are two kinds of data values that can be stored in variables, passed as arguments, returned by functions, and operated on: primitive values and reference values.

3.2.1 Primitives

The primitive types are the boolean type and the numeric type. The numeric types are the integral types `int` and `char`, and the floating point type `double`.

- `int`: a signed 32-bit integer type made up of a sequence of digits representing a number in base 10. A `-` symbol is used to denote negative numbers.
- `double`: a signed 64-bit floating point type, IEEE 754 floating point value. A `-` symbol is used to denote negative numbers.

- bool: 1 byte, 00000001 for true, 00000000 for false. Booleans will default to false unless otherwise assigned.
- string: a sequence of characters enclosed in double quotes.

3.2.2 Mutability

In GWiz, all primitive objects are mutable, including ints, doubles, strings, booleans. Trackpoint structs are also mutable. Linked Lists are immutable except for appending elements. The file type is also mutable.

3.3 Built In Types

3.3.1 Trackpoint

A Trackpoint is a built in data type used to store the latitude, longitude and time. Types must be a double, double, and int, respectively.

Declaring and Initializing Trackpoints

A Trackpoint is declared with the following syntax.

```
1 Trackpoint t;
2 t=(3.52315, 5.14243, 63425);
```

A Trackpoint can also take in defined variables as arguments.

```
1 Trackpoint t;
2 double lat;
3
4 lat=2.4;
5 t=(lat, 3.53134, 248233);
```

Accessing Trackpoint Elements

Trackpoints have 3 associated functions: `getLat()`, `getLon()`, `getTime()` to get the latitude, longitude, and time of a Trackpoint.

```
1 Trackpoint t;
2 int time;
3 t=(4.2, 6.7, 89);
4 time=getTime(t);
```

3.3.2 Linked List

GWiz fully supports strongly typed Linked Lists of type Int and Trackpoint. Lists of other types, e.g. bool, double, String, or File may be declared, but the elements cannot be accessed. The underlying implementation of Linked List is located in the C file list.c. Users must initialize a Linked List with a minimum of one element, and can append elements and access at specified indices. Linked

Lists have dynamically allocated memory.

Declaring and Initializing Linked List

A Linked List is declared using the following syntax.

```
1   Trackpoint[] a;  
2   int[] b;
```

A Linked List must be initialized with a minimum of one element to allocate memory. Upon initialization, if the List literal is empty, a compile time error will occur. If List literal contains different types, a compile time error will occur. A Linked List is initialized using the following syntax.

```
1   a = [(1.0, 2.0, 3)]; /* a is initialized with one Trackpoint */  
2   b = [1, 2, 3, 4, 5];
```

Appending Linked List

Users can append an element to the end of a Linked List. If a user tries to append to a non-Linked List type, a compile time error will occur. If a user tries to append an element of a different type than other elements in the List, a runtime error will occur.

Syntax for appending an element follows:

```
1   Trackpoint c;  
2   int d;  
3  
4   c = (2.0, 3.0, 4);  
5   d = 56;  
6  
7   a +<c>;           /* Trackpoint c is appended to list a */  
8   b +<d>;           /* int d is appended to int list b */
```

Accessing Linked List Elements

Users can access, but not modify, elements at specified indices in the Linked List. If a user tries to access an element that is not an Int or a Trackpoint, a runtime error will occur. If the user tries to access a type other than Linked List, a runtime error will occur. Accessing elements in an out-of-bounds index will cause undefined behavior

Syntax for accessing a Linked List element follows:

```
1   Trackpoint e;  
2  
3   e = a-<0>         /* Access and element at position 0 in list a  
   */
```

3.4 Scoping Rules

GWiz is a statically scoped language. Variables only exist inside of the block they were declared in. A block can be a for loop, if else statement, or while loop. Variables outside of all blocks have global scope and can be access anywhere in the program after their declaration. Multiple variables in the same scope cannot have the same name regardless of their type.

3.5 Memory

GWiz dynamically allocates and leaks memory. It does not require explicit memory allocation or management. Objects are passed by value, not by reference.

3.6 Expressions and Operators

An expression consists of at least one operand and zero or more operators. Available operators in GWiz are detailed in the following subsections.

3.6.1 Unary Operators

Unary operators act on an expression. In GWiz, NEG and NOT are the two unary operators. NEG is denoted by the symbol - and indicates the negation of an integer or float. NOT is denoted by the symbol ! and indicates the negation of a boolean expression.

3.6.2 Binary Operators

Binary operators act on two expressions. Examples of binary operators in GWiz include: +, -, *, /, =, ==, !, <, >, <=, >=, &&, ||

1. Arithmetic Operators

- (a) Addition is performed on two values of the same type. Two strings can also be concatenated using the addition operator.

```
1      2 + 3 /* Evaluates to 5 */
2      1.2 + 5.0 /* Evaluates to 6.2 */
```

- (b) Subtraction is performed on two values of the same type.

```
1      10 - 3 /* Evaluates to 7 */
2      9.2 - 5.1 /* Evaluates to 4.1 */
```

- (c) Multiplication is performed on two values of the same type.

```
1      8 * 6 /* Evaluates to 48 */
2      2.2 * 1.1 /* Evaluates to 2.42 */
```

(d) Division is performed on two values of the same type.

```
1          10 / 2 /* Evaluates to 5 */
2          3.0 / 0.5 /* Evaluates to 6.0 */
```

2. Assignment Operator

The assignment operator, denoted by =, stores a value in a variable. The variable appears on the left of the = and the value to store appears on the right.

```
1      x = 3.0; /* The value 3.0 is stored in the variable x */
```

3. Relational Operators

Relational operators determine how two operands relate to each other. In GWiz, relational operators include equal to, not equal to, greater than, and less than, denoted by ==, !=, >, <, >=, <= respectively. An expression containing two inputs and a relational operator returns true or false.

```
1      x = 0;
2      y = 1;
3      x > y /* Evaluates to false */
4      x == y /* Evaluates to false */
5      x != y /* Evaluates to true */
```

3.6.3 Operator Precedence

Operator precedence from lowest to highest precedence:

| Operator | Meaning | Associativity |
|----------|-------------------------|---------------|
| ; | Statement end | Left |
| = | Assignment | Right |
| . | Access | Left |
| | OR | Left |
| && | AND | Left |
| = != | Equality/Inequality | Left |
| ><>=<= | Comparison | Left |
| + - | Addition/Subtraction | Left |
| */ | Multiplication/Division | Left |
| ! | NOT | Right |
| - | NEGATION | Right |

3.7 Statements

In GWiz, a statement is one of the following:

- expression

- return statement
- if statement
- if else statement
- while loop
- for loop

3.7.1 If else Statement

An if statement evaluates a condition (an expression) in parentheses to be true, the program will evaluate the statements demarcated by immediate curly braces. Otherwise, if the condition evaluates to false, the program will check for an else statement and execute the block of statements contained in the curly braces following else.

```
1 if (condition) {
2     /* some statements */
3 } else {
4     /* some other statements */
5 }
```

3.7.2 For Statement

For statements are used to iterate through a range of values. The statement must include an expression that initializes a looping variable, an expression that constrains that variable to indicate when to stop looping, and an expression that increments or changes the looping variable. Statements inside the for loop as many times as the variable is incremented and/or execute with the value of the looping variable.

```
1 for (int i=0; i<10; i=i+1) {
2     /* some statements */
3 }
```

3.7.3 Return Statement

The return statement indicates the end of a function's execution and returns control to the function that called it. The type of the return value must match the return type explicitly named in the function declaration.

```
1 int foo() {
2     int x = 2;
3     return x; /* x is of type int */
4 }
```

3.8 While Statement

While statements consist of a condition and a series of statements. The statements are repeatedly evaluated as long as the condition remains true.

```
1 while (condition) {  
2     /* some statements */  
3 }
```

3.9 Functions

3.9.1 Function Declarations

A function statement takes certain inputs as parameters and returns one value. The body of a function statement is delimited by curly braces, and the return type of the function value must be explicitly stated. The type returned must match with the expected return type. An example of a function declaration follows:

```
1 int foo(int x, int y) {  
2     return x+y;  
3 }
```

3.9.2 Function Calls

A function is called by its identifier. Its arguments must be contained in parentheses and separated by commas. An example of a function call follows:

```
1 foo(2,3);
```

3.9.3 Variable assignment from functions

A function may be called as the right-hand side of a variable assignment. The variable would be assigned to the return value of the function as follows:

```
1 int x = foo(2,3);
```

3.9.4 General Functions

Printing

A call to a printing function will print an argument to standard output. Printing should be called with

1. `printi(int i)` for an integer
2. `printd(double d)` for a double
3. `prints(string s)` for a string

4. `printb(bool b)` for a bool.

The return type is void. Examples with the standard output commented follow:

```
1 printi(23);      /* 23 */
2 printd(2.0);    /* 2.0 */
3 prints("hello"); /* hello */
4 printb(true);   /* true */
```

3.9.5 Trackpoint Functions

`getLat(Trackpoint t)`

The `getLat()` function takes in a `Trackpoint` and returns a double. The double corresponds to the latitude of the `Trackpoint`.

```
1 Trackpoint t;
2 double d;
3
4 t=(3.4, 2.5, 45);
5 d=getLat(t); /* d now equals 3.4 */
```

`getLon(Trackpoint t)`

The `getLon()` function takes in a `Trackpoint` and returns a double. The double corresponds to the longitude of the `Trackpoint`.

```
1 Trackpoint t;
2 double d;
3
4 t=(3.4, 2.5, 45);
5 d=getLon(t); /* d now equals 2.5 */
```

`getTime(Trackpoint t)`

The `getTime()` function takes in a `Trackpoint` and returns an int. The int corresponds to the time of the `Trackpoint`.

```
1 Trackpoint t;
2 int i;
3
4 t=(3.4, 2.5, 45);
5 i=getTime(t); /* i now equals 45 */
```

3.9.6 GPX File Functions

`file parseGPX(string filepath)`

`parseGPX` takes in a filepath to the `.gpx` file you would like to parse as a string and returns a file type.

```
1 file f;
2 f = parseGPX("test.gpx");
```

void stat(file f)

`stat` takes in a file pointer to a parsed GPX file and prints out a quick summary of the major statistics of the file, including, total distance travelled, time elapsed, and average speed. This a sample output from the `stat` function:

```
1 ***** GPX FILE SUMMARY *****
2 Distance Covered: 65.046711 km
3 Elapsed Time: 147 min
4 Average Speed: 26.549678 km/h
5 *****
```

double totDist(file f)

`totDist` takes in a file pointer to a parsed GPX file and returns the total distance travelled in kilometers.

```
1 file f;
2 double dist;
3
4 f = parseGPX("test.gpx");
5 dist = totDist(f);
```

int totTime(file f)

`totTime` takes in a file pointer to a parsed GPX file and returns the total time elapsed in minutes.

```
1 file f;
2 int time;
3
4 f = parseGPX("test.gpx");
5 time = totTime(f);
```

double avgSpeed(file f)

`avgSpeed` takes in a file pointer to a parsed GPX file and returns the average speed in kilometers per hour.

```
1 file f;
2 double speed;
3
4 f = parseGPX("test.gpx");
5 speed = avgSpeed(f);
```

4 Project Plan

4.1 Planning, Specification, Development and Testing

Team meetings occurred (virtually) 1-2 times each week throughout the semester, typically on Thursday afternoons to utilize Evan's office hours later in the night and on Sundays to create a plan with deliverables for the week. We increased

meeting frequency around large external and internal deadlines. Aside from our scheduled Google-Meet meetings, we corresponded regularly over text. We also scheduled "hackathons" where we would have an open Google-Meet for team members to drop into for debugging help or to work together.

At the beginning of the semester, we focused on brainstorming ways to parse GPX files so we could access the data in GWiz. Once we had a preliminary idea on how to do this, we divided up parts of the project and assigned them to individuals or to pairs. Rebecca and Elisa focused on parsing and linking the GPX data, and Ashley and Katherine focused on implementing the syntax, semantic checking, and code generation for non-explicitly GPX related types and operations.

After helloworld, we started adding in our built-in types. Our meeting time was then focused on ensuring the parsed GPX file would line up with the built-ins in Gwiz. Properly accessing GPX data is a key feature of our language.

After we were able to generate code and link to a GPX file, we created a more robust testing suite that congregated all of the tests that had been used for each aspect of the project, and added new ones. Individual members of the team were responsible for creating passing and failing test units for parts that they implemented.

4.2 Style Guide

We followed the following style guidelines while developing our compiler in order to ensure consistency, readability, and transparency:

- OCaml editing and formatting style to write code for compiler architecture.
- C language editing and formatting style to write code for linked C functions.

Additional style guidelines for writing code in GWiz.

- File names end in .gw
- Use indentation to clearly indicate dependencies and statement blocks.
- Variable identifiers begin with a lowercase letter and are camelcase.
- Function identifiers begin with a lowercase letter and are camelcase.

4.3 Project Timeline

| Milestone | Date |
|---------------------------------|------|
| LRM, Parser | 2/24 |
| Hello World | 3/21 |
| Parsed GPX Data | 4/5 |
| Linked List and Trackpoint | 4/15 |
| Obtain GPX Data in gWiz | 4/22 |
| Built in Functions for GPX data | 4/25 |

4.4 Roles and Responsibilities

| Team Member | Responsibility |
|-------------|---|
| Ashley | Compiler front end, semantics, code generation, test units and automation |
| Elisa | Code generation, link to parsed GPX file, built-ins |
| Katherine | Compiler front end, semantics, code generation |
| Rebecca | GPX parsing, test units and automation |

4.5 Software Development Environment

- Programming language for building the compiler: OCaml version 4.05.0. Ocaml yacc and Ocamllex extensions were used for compiling the scanner and parser.
- Development environments: vim, Visual Studio Code,
- Software: A git repository on GitHub was used for version control. LaTeX was used to create the project proposal, language reference manual and the final report.
- C libraries: stdio.h, stdlib.h, string.h, time.h, math.h, libxml/xmlmemory.h, libxml/parser.h, libxml/xpath.h, libxml/xpathInternals.h

5 Language Evolution

The primary focus of GWiz is to be able to easily interpret and analyze GPX files. While this motivation has remained constant throughout the semester, how we approached implementing solutions has undergone several iterations.

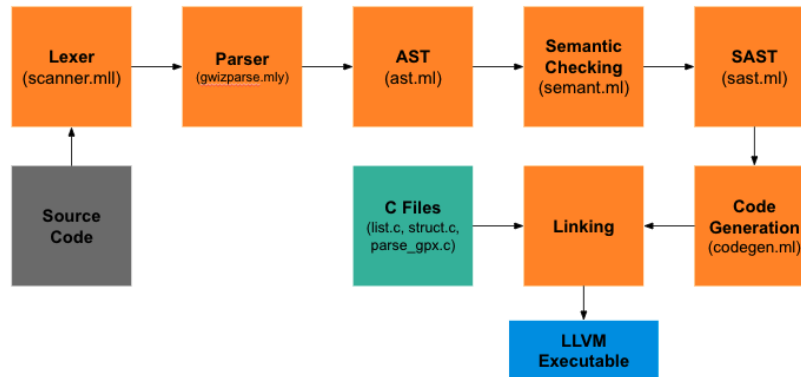
At the beginning, we envisioned GWiz to be an object-oriented Java-like language, with classes and constructors. Most notably, we intended for GWiz to have an Activity class built into its standard library. Consequently, each GPX file could be represented as an Activity type, constructed of multiple different classes, through the structure of linked list. However, it became clear that this implementation was too cumbersome for the user, and was incredibly inefficient.

The second iteration of GWiz was envisioned to be where a parsed GPX file was stored in a C array of structs, which was then passed to GWiz for the user to access. However, we were not able to create an array of structs where the memory allocations matched up with C.

This led us to the third and final iteration of GWiz. In this implementation, we decided to keep all the data from the parsed GPX file as a c array of structs (trackpoints) and simply pass this pointer directly to GWiz. We created a `file` type in GWiz to handle this "file pointer" (which under the hood pointed to the parsed GPX data neatly stored away in an array rather than the actual file itself). This meant that the user did not have to deal with all the extra junk in the XML file directly, and could simply pass the pointer into the built-in GPX functions. It is also incredibly simple to initialize a `file` in GWiz, making it easy to work with multiple GPX files in one program. Finally, this implementation is more efficient in terms of memory as we don't have to copy over any memory from C, so we'll run a lower risk of running out of memory when working with many and/or large GPX files.

6 Architectural Design

6.1 The Compiler



6.2 The Lexer

The lexer (`scanner.mll`) takes in a Gwiz source program and generates tokens for identifiers, keywords, operators, and values. This was implemented by Ashley and Katherine.

6.3 The Parser

The parser (`gwizparse.mly`) converts the tokens from the Lexer into an abstract syntax tree (AST), based on the context free grammar rules. This was implemented by Ashley, Elisa, and Katherine.

6.4 Semantic Checking

The semantic checker (`semant.ml`) recursively traverses the AST and converts it into a semantically checked abstract syntax Tree (SAST). This was implemented by Ashley and Katherine.

6.5 Code Generation

The code generator (`codegen.ml`) traverses the semantically checked abstract syntax tree (SAST) to post order generate code. This was implemented by Ashley, Elisa, and Katherine.

7 C Libraries

7.1 `parse_gpx.c`

`parse_gpx.c` contains the function `parseGPX()`, in addition to functions that help process trackpoint data. `parseGPX()` takes in a file path, parses a GPX file using `libxml2`, and constructs an array of trackpoints. It then returns a pointer to the array. Other functions included are `totDis()`, `totTime()`, `avgSpeed()`, `stat()`, and helper functions. This was implemented by Elisa and Rebecca.

7.2 `List.c`

`list.c` provides the implementation of the Linked List type. Elements are stored as structs, with a data attribute which contains the 64-bit unsigned long cast of the element's pointer and a pointer to the next element in the list. `List.c` contains functions to initialize a list, append a node to the end, and to access an element. There is no memory management, and the List is stored on the heap. This was implemented by Katherine.

7.3 `struct.c`

`Struct.c` and its associated header file, `struct.h`, contains the structure of the Trackpoint type in `gwiz`. The main object is `struct Trackpoint` which contains a double, double, and int. The built in `getLat()`, `getLon()` and `getTime()` functions are also declared here. This was implemented by Ashley.

8 Test Plan

We tested our project in two waves. The first round of testing was designed mainly to test that we could scan & parse tokens and construct the expected abstract syntax tree. This is described in more detail under the Unit Testing subsection. The second round of testing occurred once we were able to generate code. This round is described in more detail under the Integration Tests & Test Automation subsections.

8.1 Unit Testing

We tested our Scanner and Parser using `menhir` to construct a CST. All of the testing for this round was done in the command line, so no files are included at the end. Key features tested were variable declarations, function declarations, and nested expressions.

8.2 Integration Tests, Test Automation and Scripts

Once we were able to generate code, we began building a test suite. Members of our team were expected to contribute units to the test suite for features that they worked on once the feature was completed. We then split up the remaining features and contributed tests for those features. Ashley and Rebecca contributed test cases for operators, variable declarations, primitive data types, and printing. Elisa contributed tests for all GPX built-ins and function declarations. Katherine contributed Linked List test cases. These units were tested using two scripts, `printout.sh` which simply compiled and ran an individual unit for easier debugging, and `testall.sh`. `Testall.sh` compares the output of each unit to the expected output, echoing OK if the correct stdout or error message is produced. `Testall.sh` can be run with the command `"make test"`.

8.3 Example Test Programs

tests/test-gpx3.gw This test program was chosen because it includes parsing a GPX file and calling a built-in function on that parsed GPX file. These are key features of GWiz.

```
1 int main(){
2     file f;
3     f = parseGPX("test.gpx");
4     stat(f);
5     return 0;
6
7 }
```

LLVM Code - `tests/test-gpx3.gw`

```
; ModuleID = 'gWiz'
source_filename = "gWiz"
```

```

%Array = type { %Node*, i32 }
%Node = type { i64, %Node* }
%trackpoint_t = type { double, double, i32 }

@fmt = private unnamed_addr constant [4 x i8] c"%d\0A\00"
@fmt.1 = private unnamed_addr constant [4 x i8] c"%s\0A\00"
@fmt.2 = private unnamed_addr constant [4 x i8] c"%g\0A\00"
@fmt.3 = private unnamed_addr constant [4 x i8] c"%d\0A\00"
@str = private unnamed_addr constant [9 x i8] c"test.gpx\00"

declare i32 @printf(i8*, ...)

declare %Array* @initList(i64, ...)

declare %Node* @addEnd(%Array*, i64, ...)

declare i64 @getElemIndex(%Array*, i32, ...)

declare i32 @numPt(i8*)

declare i32* @parseGPX(i8*)

declare i32 @printFirst(i32*)

declare i32 @totTime(i32*)

declare double @totDist(i32*)

declare double @avgSpeed(i32*)

declare i32 @stat(i32*)

declare i32 @intFloor(double)

declare double @squareRoot(double)

declare double @doubleFloor(double)

declare double @getLat(%trackpoint_t*)

declare double @getLon(%trackpoint_t*)

declare i32 @getTime(%trackpoint_t*)

```

```

define i32 @main() {
entry:
  %f = alloca i32*
  %parseGPX = call i32* @parseGPX(i8* getelementptr inbounds ([9 x i8],
    [9 x i8]* @str, i32 0, i32 0))
  store i32* %parseGPX, i32** %f
  %f1 = load i32*, i32** %f
  %stat = call i32 @stat(i32* %f1)
  ret i32 0
}

```

tests/test-array1.gw This test program was chosen because it includes Linked Lists, Trackpoint structs, and looping. These features are key for users to define their own routes.

```

1  int main() {
2      Trackpoint[] a;
3      Trackpoint b;
4      Trackpoint c;
5      Trackpoint d;
6      Trackpoint e;
7      int time;
8      int i;
9
10     b = (1.0, 2.0, 3);
11     c = (1.5, 2.9, 8);
12     d = (2.8, 89.1, 87);
13
14     a = [b];
15     a +<c>;
16     a +<d>;
17
18     for (i = 0; i < 3; i = i + 1) {
19         e = a-<i>;
20         time = getTime(e);
21         printi(time);
22     }
23     return 0;
24 }

```

LLVM Code - tests/test-array1.gw

```

; ModuleID = 'gWiz'
source_filename = "gWiz"

%Array = type { %Node*, i32 }
%Node = type { i64, %Node* }
%trackpoint_t = type { double, double, i32 }

```

```

@fmt = private unnamed_addr constant [4 x i8] c"%d\0A\00"
@fmt.1 = private unnamed_addr constant [4 x i8] c"%s\0A\00"
@fmt.2 = private unnamed_addr constant [4 x i8] c"%g\0A\00"
@fmt.3 = private unnamed_addr constant [4 x i8] c"%d\0A\00"

declare i32 @printf(i8*, ...)

declare %Array* @initList(i64, ...)

declare %Node* @addEnd(%Array*, i64, ...)

declare i64 @getElemIndex(%Array*, i32, ...)

declare i32 @numPt(i8*)

declare i32* @parseGPX(i8*)

declare i32 @printFirst(i32*)

declare i32 @totTime(i32*)

declare double @totDist(i32*)

declare double @avgSpeed(i32*)

declare i32 @stat(i32*)

declare i32 @intFloor(double)

declare double @squareRoot(double)

declare double @doubleFloor(double)

declare double @getLat(%trackpoint_t*)

declare double @getLon(%trackpoint_t*)

declare i32 @getTime(%trackpoint_t*)

define i32 @main() {
entry:
    %a = alloca %Array*
    %b = alloca %trackpoint_t*
    %c = alloca %trackpoint_t*
    %d = alloca %trackpoint_t*

```

```

%e = alloca %trackpoint_t*
%time = alloca i32
%i = alloca i32
%trackpoint_tmp = alloca %trackpoint_t
%trackpoint_pointer = alloca %trackpoint_t*
%lat = getelementptr inbounds %trackpoint_t, %trackpoint_t* %trackpoint_tmp,
i32 0, i32 0
store double 1.000000e+00, double* %lat
%lon = getelementptr inbounds %trackpoint_t, %trackpoint_t* %trackpoint_tmp,
i32 0, i32 1
store double 2.000000e+00, double* %lon
%time1 = getelementptr inbounds %trackpoint_t, %trackpoint_t* %trackpoint_tmp,
i32 0, i32 2
store i32 3, i32* %time1
store %trackpoint_t* %trackpoint_tmp, %trackpoint_t** %trackpoint_pointer
%0 = load %trackpoint_t*, %trackpoint_t** %trackpoint_pointer
store %trackpoint_t* %0, %trackpoint_t** %b
%trackpoint_tmp2 = alloca %trackpoint_t
%trackpoint_pointer3 = alloca %trackpoint_t*
%lat4 = getelementptr inbounds %trackpoint_t, %trackpoint_t* %trackpoint_tmp2,
i32 0, i32 0
store double 1.500000e+00, double* %lat4
%lon5 = getelementptr inbounds %trackpoint_t, %trackpoint_t* %trackpoint_tmp2,
i32 0, i32 1
store double 2.900000e+00, double* %lon5
%time6 = getelementptr inbounds %trackpoint_t, %trackpoint_t* %trackpoint_tmp2,
i32 0, i32 2
store i32 8, i32* %time6
store %trackpoint_t* %trackpoint_tmp2, %trackpoint_t** %trackpoint_pointer3
%1 = load %trackpoint_t*, %trackpoint_t** %trackpoint_pointer3
store %trackpoint_t* %1, %trackpoint_t** %c
%trackpoint_tmp7 = alloca %trackpoint_t
%trackpoint_pointer8 = alloca %trackpoint_t*
%lat9 = getelementptr inbounds %trackpoint_t, %trackpoint_t* %trackpoint_tmp7,
i32 0, i32 0
store double 2.800000e+00, double* %lat9
%lon10 = getelementptr inbounds %trackpoint_t, %trackpoint_t* %trackpoint_tmp7,
i32 0, i32 1
store double 8.910000e+01, double* %lon10
%time11 = getelementptr inbounds %trackpoint_t, %trackpoint_t* %trackpoint_tmp7,
i32 0, i32 2
store i32 87, i32* %time11
store %trackpoint_t* %trackpoint_tmp7, %trackpoint_t** %trackpoint_pointer8
%2 = load %trackpoint_t*, %trackpoint_t** %trackpoint_pointer8
store %trackpoint_t* %2, %trackpoint_t** %d
%initList = call %Array* (i64, ...) @initList(i64 ptrtoint (i1** getelementptr

```



```

(i1*, i1** null, i32 1) to i64))
%b12 = load %trackpoint_t*, %trackpoint_t** %b
%3 = ptrtoint %trackpoint_t* %b12 to i64
%addEnd = call %Node* (%Array*, i64, ...) @addEnd(%Array* %initList, i64 %3)
store %Array* %initList, %Array** %a
%c13 = load %trackpoint_t*, %trackpoint_t** %c
%4 = ptrtoint %trackpoint_t* %c13 to i64
%a14 = load %Array*, %Array** %a
%addEnd15 = call %Node* (%Array*, i64, ...) @addEnd(%Array* %a14, i64 %4)
%d16 = load %trackpoint_t*, %trackpoint_t** %d
%5 = ptrtoint %trackpoint_t* %d16 to i64
%a17 = load %Array*, %Array** %a
%addEnd18 = call %Node* (%Array*, i64, ...) @addEnd(%Array* %a17, i64 %5)
store i32 0, i32* %i
br label %while

while:                                     ; preds = %while_body, %entry
%i24 = load i32, i32* %i
%tmp25 = icmp slt i32 %i24, 3
br i1 %tmp25, label %while_body, label %merge

while_body:                               ; preds = %while
%i19 = load i32, i32* %i
%a20 = load %Array*, %Array** %a
%getElement = call i64 (%Array*, i32, ...) @getElement(%Array* %a20, i32 %i19)
%arrayAcc = inttoptr i64 %getElement to %trackpoint_t*
store %trackpoint_t* %arrayAcc, %trackpoint_t** %e
%e21 = load %trackpoint_t*, %trackpoint_t** %e
%getTime = call i32 @getTime(%trackpoint_t* %e21)
store i32 %getTime, i32* %time
%time22 = load i32, i32* %time
%printf = call i32 (i8*, ...) @printf(i8* getelementptr inbounds ([4 x i8],
[4 x i8]* @fmt, i32 0, i32 0), i32 %time22)
%i23 = load i32, i32* %i
%tmp = add i32 %i23, 1
store i32 %tmp, i32* %i
br label %while

merge:                                     ; preds = %while
ret i32 0
}

```

9 Lessons Learned

Ashley

I will be honest, this was a hard project. In the early stages, we had this nebulous big picture idea but no idea how to go about actually implementing it. We went through a lot of brainstorming and trial and error to see what ideas were actually feasible in the period of time given. I learned how to think on both a large and small scale. We had to not only figure out how to actually create a language that can do basic scoping, assignment and arithmetic but also how to get a gpx file parsed and into gwiz for analysis.

This brings me to the topic of the big picture. I learned every small idea or element has to fit in with the larger goal of the project since there is a huge ripple effect. If certain elements aren't necessary to your language, don't implement them. If they are cool but not directly related to what you are trying to achieve, make it a low priority. It's not a bad idea to have a bucket load of built-in ideas early on, but know that you will need to be flexible. A good mantra would be have a plan but stretch because you will need to be flexible. And because sitting in front of the computer for hours on end will give you tight shoulders and a sore neck.

Overall though, I learned a lot. This class brought together concepts from CS Theory and AP which was really cool. I got to learn how to use git for a group project and also work on a programming project for longer than a 2 week time period. We all knew each other pre-pandemic which made it easier since there was no icebreaker period.

Definitely start as early as you can. As soon as you see the words ocamlyacc, get working. We found it easiest to divide up built ins and different parts of the larger process. It might take a lot of trial and error, office hours and late nights but once you get your first cool program working, it's all almost worth it.

Elisa

Oh boy, what a journey this project was.

This was my first "big girl" coding project. It was great getting the experience of working through all the stages of development - from the idea phase to testing the final product it's fun seeing your project come to life.

I won't lie - the process of learning the tools needed to make this project happen was painful. But you know what they say, right? No pain, no gain.

Aside from coding, project management and collaboration skills are great takeaways from this project. When faced with a difficult, seeming insurmountable task (which is definitely what it felt like at the beginning of this project), I found that breaking the task down to certain milestones helped us think about the process in more manageable chunks. For example, one of our major milestones was to have every feature in Microc implemented correctly in GWiz, and the next one being able to store a parsed GPX file in GWiz. I'd encourage groups to think carefully about the priority of certain tasks, and start tasks that other parts of the project will depend on in the future earlier than you think you need to prevent bottle-necking.

Also, it is very important for all members of the group to get familiar with each file. For example, I did not work on `semant.ml` up until I needed to add a built-in function. To my frustration, it always returned void despite checking everything in codegen. Turns out that all built in functions were set to return void in `semant.ml`. Oops. Stuff like this is why it's important that all group members are part of the whole process. It seemed really annoying and time consuming to begin with, but it's definitely worth the rage you'll save in the long run.

Katherine

This project was my first team coding project, and my first project that required the development of a testing program—as Elisa says, a "big girl" project. I learned a lot about the importance of staying in constant contact with my teammates, especially in a project such as ours where bottle-necking can create a chain of consequences.

I also belatedly learned how essential it is to reach out for help earlier rather than later. My initial task was to implement an array. After struggling and restarting from scratch at least 4 times, I chose to switch to a Linked List implementation which I knew I could succeed in creating. However, creating an array was key to the second iteration of our development, so I tried again. The commented out code for `TrackArray` was the result of my second round of attempts—the code successfully created an array, but unfortunately it always created an array of arrays instead of an array of types. After much testing and research on LLVM on my own, I could not figure out why this was happening, but it was too late to ask for help as my team had moved on to a different iteration of the language (file pointers). Had I reached out to teammates or TAs in early April when I knew this would be a big challenge for me, I believe I would have succeeded. It is definitely frustrating to spend so much time on something that gets almost there, but I am thankful for the opportunity to learn. This project taught me more than anything else I did academically this year!

Through this experience of writing an array from scratch multiple different times, I learned so much about the amazing features of OCaml, and about code generation and LLVM. I would now consider myself a big fan of functional languages.

My advice for future students would be to try and individually implement a feature that is not directly related to anything in MicroC from the Scanner to the codegen. It is truly a great learning experience!

Rebecca

This was my first experience working on both a semester-long coding project and a team programming assignment. I enjoyed creatively problem-solving with my teammates, and also learned the importance of communication when building a project with so many components. I would advise future teams to meet early and often, and to be open to changing the plan; I think willingness to change and adapt allowed us to complete a project we are really proud of! I

also learned to appreciate Git as a collaborative tool more. I had previously used GitHub for smaller individual projects, but found it even more useful when co-developing a project with others.

Tasked with figuring out how to parse a GPX file in C, I also learned a lot in building a GPX parser with an external XML parser. It was great to implement the parser in a language I was familiar with, and also to figure out how it connects to our language and compiler. As my first experience learning a functional language, I found the learning curve for OCaml to be extremely steep, though it was cool to see how it allowed us to do more in less lines of code. It was most rewarding to see how all the pieces fell together to allow us to analyze GPX data in GWiz. All in all, I gained a lot from this project.

10 Appendix

10.1 Project Log

```
commit 2faec1656ec96b7dedf66432ff42a330697c75a9
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Mon Apr 26 23:18:08 2021 -0400
```

file cleanup

```
commit 4def20ef5213ec5d202bb1ae617681b238a2e5e4
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Mon Apr 26 23:16:58 2021 -0400
```

Delete test-array2.ll

```
commit 2dc662987248cde14bedbcee868527dcee86da9d
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Mon Apr 26 23:16:52 2021 -0400
```

Delete test-array1.ll

```
commit 8181db96bb4fe01960368dbbcc1714d42111fe80
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Mon Apr 26 23:16:44 2021 -0400
```

Delete test-array1.exe

```
commit 9d65852ced06e15da25126084f60c77c0d2c20fb
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Mon Apr 26 23:16:20 2021 -0400
```

Delete test-helloworld.s

commit ce8ae45354c019405f5432695d42f61b907ba3e2
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Mon Apr 26 23:15:39 2021 -0400

Delete gwiz.native

commit 5eb5a5bfff077bb6e9766599087546da251a5513a
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Mon Apr 26 23:15:21 2021 -0400

Delete _build directory

commit 83cd72a16ebb22e1d82083167d7efcfe410b2591
Merge: a73e863 7c16049
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Mon Apr 26 23:14:10 2021 -0400

Merge branch 'main' of <https://github.com/ashleytkim/gwiz> into main

commit a73e86364cc6a070bf6cdf5ebc2f16d232674fd6
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Mon Apr 26 23:12:48 2021 -0400

deleted array file

commit 7c16049938f1e5eb562032b9527d1d8e6d29cf3c
Author: Rebecca Yao <rebeccayao8@gmail.com>
Date: Mon Apr 26 23:11:29 2021 -0400

Added authorship

commit 5cd1a2242becc2142bf35c5dfdcceb036187c0b1
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Mon Apr 26 21:48:20 2021 -0400

updated array tests

commit aa1083785c5b19bb1760e1d887356c84fd73e07e
Merge: 691f287 53b808d
Author: kpd2128 <kpd2128@barnard.edu>
Date: Mon Apr 26 19:19:30 2021 +0000

final fail tests

commit 53b808de5d2841167707f80e3a443904696c5471
Author: Elisa Luo <eyl2130@columbia.edu>
Date: Mon Apr 26 14:31:08 2021 -0700

added gpx files for testing

commit 2ecddc6f689700205c21fa4fca8c3d203f43919d
Author: Elisa Luo <eyl2130@columbia.edu>
Date: Mon Apr 26 14:30:11 2021 -0700

added code demo files

commit a00a776b2d37c8281e29c0bdd7d94e11ce6b987f
Author: Elisa Luo <eyl2130@columbia.edu>
Date: Mon Apr 26 14:28:41 2021 -0700

fixed ast warning

commit 691f287371e88a89de1437121463d7e1542baa51
Author: kpd2128 <kpd2128@barnard.edu>
Date: Mon Apr 26 19:19:00 2021 +0000

final fail tests

commit facefc74ffeee278d32f2cb900d6e1e8de38f856
Merge: 26b1c18 1d0baba
Author: kpd2128 <kpd2128@barnard.edu>
Date: Mon Apr 26 15:22:56 2021 +0000

remove warning

commit 1d0baba4b841eedb50b60e5b1bf6160647b828f4
Author: Elisa Luo <eyl2130@columbia.edu>
Date: Mon Apr 26 14:11:22 2021 -0700

delete smth from semant

commit 14a816111e780e076cb3c32dedc2b8541b16c1b2
Author: Elisa Luo <eyl2130@columbia.edu>
Date: Mon Apr 26 14:08:35 2021 -0700

add .out files for math and pace

commit 3be6c4ee23796c40d02b928ee963a7b4827bd768
Author: Elisa Luo <eyl2130@columbia.edu>
Date: Mon Apr 26 14:03:37 2021 -0700

added doubleFloor

commit 12edec1cf773ffcd50def874828908536e4461a8
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Mon Apr 26 13:04:45 2021 -0400

updated list demo

commit 26b1c18eced969574c91e902b0d594b4261869e6

Author: kpd2128 <kpd2128@barnard.edu>
Date: Mon Apr 26 15:21:48 2021 +0000

remove warning

commit c4d8ebedca0cb0dadd7025a79f06822d0c441209
Author: kpd2128 <kpd2128@barnard.edu>
Date: Mon Apr 26 06:44:32 2021 -0400

updated with square root!

commit 65f341b986951640bcefb02fbb12152066896308
Author: Elisa Luo <eyl2130@columbia.edu>
Date: Sun Apr 25 21:03:34 2021 -0700

added square root and floor

commit 3d850241e549d4618b626766b7cb7f6dbe3aac4b
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Sun Apr 25 23:11:56 2021 -0400

Delete gwiz.native

commit 3c557bff9f94be5b3ac7d6956a221c7f3c011b85
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sun Apr 25 23:10:54 2021 -0400

updated array tests

commit 4ec8e3478c3115cfc80f4a4c7a86283839633091
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sun Apr 25 22:48:37 2021 -0400

no longer exponential time

commit 00ab5e0dbd9becd1bed5ad128ef5d13163aa8278
Merge: 79b82a2 56342f0
Author: Rebecca Yao <rebeccayao8@gmail.com>
Date: Sun Apr 25 22:02:52 2021 -0400

Merge branch 'main' of <https://github.com/ashleytkim/gwiz> into main

commit 79b82a2e29be9d9df3c9e52edca4618d1f965f4d
Author: Rebecca Yao <rebeccayao8@gmail.com>
Date: Sun Apr 25 22:02:44 2021 -0400

Edited listdemo.gw

commit 56342f03ad920f5d70a3d84aba1e339472c3cba1
Author: kpd2128 <kpd2128@barnard.edu>

Date: Sun Apr 25 19:57:09 2021 +0000

updated demo

commit eff28cc49c3ea82d404527035a976d3462526562
Author: kpd2128 <kpd2128@barnard.edu>
Date: Sun Apr 25 19:56:13 2021 +0000

updated demo

commit a04e3e55b0ee0b20c846d9213af6b3a570d5bd83
Author: kpd2128 <kpd2128@barnard.edu>
Date: Sun Apr 25 19:29:29 2021 +0000

more tests and demo, although I still can't compile

commit 4e3dc739244ab26ba9b77b8656b50bf6ef324d08
Author: Rebecca Yao <rebeccayao8@gmail.com>
Date: Sun Apr 25 19:14:07 2021 -0400

Updating test-array files

commit 9a3830e8bf96ae684faa134727a944ba4fb029e2
Merge: f6b319f 630eb29
Author: kpd2128 <kpd2128@barnard.edu>
Date: Sun Apr 25 17:06:25 2021 +0000

updated test 1 and 2

commit 630eb29db4a10a70fe699adcb693b13514fc55a4
Merge: d55e7a9 752cf70
Author: Rebecca Yao <rebeccayao8@gmail.com>
Date: Sun Apr 25 18:47:05 2021 -0400

Merge branch 'main' of <https://github.com/ashleytkim/gwiz> into main

commit d55e7a9dfac6ad696b9ec2fa1a1f05845cfac329
Author: Rebecca Yao <rebeccayao8@gmail.com>
Date: Sun Apr 25 18:46:56 2021 -0400

Deleted test-arrayaccess and test-arrayliteral

commit 752cf7068aab2727aa73d8ded253f9f284049de2
Merge: c0fd314 114838a
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sun Apr 25 18:46:14 2021 -0400

Merge branch 'main' of <https://github.com/ashleytkim/gwiz> into main

commit c0fd314dc3f093b3a4554c0acbaea46aeec5e714

Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sun Apr 25 18:46:09 2021 -0400

cleanup up trackpoint

commit 114838ab159da7863f42929fc2235c1dca580b1b
Merge: cc2fe11 8007fbc
Author: Rebecca Yao <rebeccayao8@gmail.com>
Date: Sun Apr 25 18:43:57 2021 -0400

Modified printall.sh file

commit cc2fe11dd6103ab141606d216c3e43877616aff1
Author: Rebecca Yao <rebeccayao8@gmail.com>
Date: Sun Apr 25 18:42:53 2021 -0400

Added test-array files

commit 8007fbca620b3493978dd08f95f0141b6bd8f03f
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sun Apr 25 18:41:38 2021 -0400

fixed gpx tests

commit 2efdb88f68917ae764720fc1c214c9687c77d070
Merge: f81f243 c8fc5a4
Author: Rebecca Yao <rebeccayao8@gmail.com>
Date: Sun Apr 25 18:27:09 2021 -0400

Merge branch 'main' of <https://github.com/ashleytkim/gwiz> into main

commit f81f24316dc13b4afe8457210e40a157b5ca9070
Author: Rebecca Yao <rebeccayao8@gmail.com>
Date: Sun Apr 25 18:26:47 2021 -0400

Modifying array test

commit c8fc5a4617060b4c71a8f8d9a93e3bcbe8314f37
Author: Elisa Luo <eyl2130@columbia.edu>
Date: Sun Apr 25 15:23:23 2021 -0700

added tests for gpx

commit eb879ad8b55fd0e9f16893a2c45250b1d1c91da8
Author: Elisa Luo <eyl2130@columbia.edu>
Date: Sun Apr 25 15:13:36 2021 -0700

update gwiz compiler

commit 5542ac06bd869abe94d2b6fe72177e354ff8711b

Merge: 2a1fccb a1c6ae7
Author: Elisa Luo <eyl2130@columbia.edu>
Date: Sun Apr 25 15:08:55 2021 -0700

Merge branch 'main' of <https://github.com/ashleytkim/gwiz> into main

commit 2a1fccb7fd6ad6d7d3d9774e95ba074760ab1221
Author: Elisa Luo <eyl2130@columbia.edu>
Date: Sun Apr 25 15:08:48 2021 -0700

deleted parsegpx from makefile and shell script

commit 77532711b573b9afc581441cc6fbc347bd14299a
Author: Elisa Luo <eyl2130@columbia.edu>
Date: Sun Apr 25 15:02:10 2021 -0700

removed parsegpx test function

commit ff603700878fa1c9b84e0f5306d7d981e29b8a2a
Merge: d7f77d3 0fb8dfd
Author: Elisa Luo <eyl2130@columbia.edu>
Date: Sun Apr 25 14:55:56 2021 -0700

Merge branch 'main' of <https://github.com/ashleytkim/gwiz> into main

commit d7f77d3af73e7219be62dfd54077da7e03fdea35
Author: Elisa Luo <eyl2130@columbia.edu>
Date: Sun Apr 25 14:55:46 2021 -0700

added shell script to compile one file

commit f6b319f59dde1b4a89e0cc89f4852724c5f4e898
Author: kpd2128 <kpd2128@barnard.edu>
Date: Sun Apr 25 17:01:28 2021 +0000

updated test 1 and 2

commit a1c6ae719869c4a06e48205dbea6bceaab901f6c
Merge: 0dfd30f 7753271
Author: kpd2128 <kpd2128@barnard.edu>
Date: Sun Apr 25 16:02:41 2021 +0000

syntax update

commit 0dfd30f6c071e2609e13d767bdc6bdc6941d8849
Author: kpd2128 <kpd2128@barnard.edu>
Date: Sun Apr 25 15:59:47 2021 +0000

syntax update

commit 0fb8dfdacca12f3c17a78fb69579d1ec60ef5976
Author: kpd2128 <kpd2128@barnard.edu>
Date: Sun Apr 25 15:55:47 2021 +0000

two test for rebecca

commit fc2282ede5e61c4f4229e70d5adcf00e19e7ed93
Merge: 8dcc2bb 215193b
Author: kpd2128 <kpd2128@barnard.edu>
Date: Sun Apr 25 14:19:38 2021 +0000

fixed shift reduce

commit 215193b5935e2499fd7071c93105b0ef1106bf17
Merge: b596d79 c0b24aa
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sun Apr 25 16:18:45 2021 -0400

Merge branch 'main' of <https://github.com/ashleytkim/gwiz> into main

commit b596d79df9667eb27271845e4835053a92d9c826
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sun Apr 25 16:18:38 2021 -0400

added more tests

commit f90bbe3316747c2006c73f6ba092be853b058440
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Sun Apr 25 14:35:38 2021 -0400

Delete struct

commit 8dcc2bb242b2228259227ef0f5e0700ae8facbee
Author: kpd2128 <kpd2128@barnard.edu>
Date: Sun Apr 25 14:19:06 2021 +0000

fixed shift reduce

commit c0b24aaa1efe3fdc8988f23070bfd4e4eb91c029
Merge: 9c8d019 f90bbe3
Author: kpd2128 <kpd2128@barnard.edu>
Date: Sun Apr 25 13:12:02 2021 +0000

removed most warnings

commit 9c8d019c5c90e1a11d49c79f485c6f86bc21ffa6
Author: kpd2128 <kpd2128@barnard.edu>
Date: Sun Apr 25 13:11:32 2021 +0000

removed most warnings

commit cbddd82832b96da0ba451f35040fd8aeaf712a17
Author: kpd2128 <kpd2128@barnard.edu>
Date: Sun Apr 25 12:21:43 2021 +0000

with some file stuff

commit d6300a218ae71d9992abf7e61e9cc0884a782dcb
Author: kpd2128 <kpd2128@barnard.edu>
Date: Sun Apr 25 05:18:36 2021 +0000

TrackArray is officially removed, rest in peace!

commit 82bf77f483cc71be4ca9673149bacd56f80b02ba
Author: kpd2128 <kpd2128@barnard.edu>
Date: Sun Apr 25 05:03:02 2021 +0000

remove TrackArray

commit a60da1efb68ffd00697bbb2a5ece1c594c120952
Merge: 6509b7f 0c7e943
Author: kpd2128 <kpd2128@barnard.edu>
Date: Sun Apr 25 04:56:00 2021 +0000

had to commit again with my permission issues

commit 6509b7f78f7fa273d3c554e014dd5bcfad15d634
Author: kpd2128 <kpd2128@barnard.edu>
Date: Sun Apr 25 04:55:13 2021 +0000

had to commit again with my permission issues

commit 0c7e943e8386701957974ed53a0f0b35753559b4
Merge: dda10f2 7aeadfa
Author: Elisa Luo <eyl2130@columbia.edu>
Date: Sat Apr 24 21:52:38 2021 -0700

finished built in gpx functions and added tests

commit dda10f29f70d2f70a80f8df599f7f41a78b459d9
Author: Elisa Luo <eyl2130@columbia.edu>
Date: Sat Apr 24 21:50:14 2021 -0700

finished built in gpx functions and added test files for them

commit 7aeadfac2628393bff0efafad6e1187795646f26
Merge: eaee35b a690acf
Author: kpd2128 <kpd2128@barnard.edu>
Date: Sun Apr 25 04:48:17 2021 +0000

```
    committing before removing TrackArray and buildList :(

commit eae35bbc026da86e756b0b0a7ed5a8aee231849
Merge: 21fa320 78f8346
Author: kpd2128 <kpd2128@barnard.edu>
Date: Sun Apr 25 04:47:34 2021 +0000

    committing before removing TrackArray and buildList :(

commit a690acf6722bc5d05e96180eea6e780f7daf80c7
Author: Rebecca Yao <rebeccayao8@gmail.com>
Date: Sun Apr 25 00:46:35 2021 -0400

    Adding tests

commit 21fa320a6258089f9b055b2f7831848764c5e4cd
Author: kpd2128 <kpd2128@barnard.edu>
Date: Sun Apr 25 04:38:53 2021 +0000

    committing before I remove TrackArray type

commit 78f8346c8f8a15e856c53c979e380fc4a5d829a3
Merge: e8b4bdf 9c44a54
Author: Rebecca Yao <rebeccayao8@gmail.com>
Date: Sat Apr 24 23:42:04 2021 -0400

    Merge branch 'main' of https://github.com/ashleytkim/gwiz into main

commit e8b4bdf57a3422dd16f35743312fb3104d09f6ff
Author: Rebecca Yao <rebeccayao8@gmail.com>
Date: Sat Apr 24 23:41:51 2021 -0400

    Adding more tests

commit 9c44a543b79110fe06eaecd19d5cacaca1e68b0f
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 24 23:32:06 2021 -0400

    reordered tests

commit 83ea6bef45a9712dda56a770f857939874eb130c
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 24 23:29:07 2021 -0400

    Delete .DS_Store

commit 67fed349c38461358d8a95cc5645891e09aeed0a
Merge: 22e176e e8fbbbf
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 24 23:27:57 2021 -0400
```

Merge branch 'main' of <https://github.com/ashleytkim/gwiz> into main

commit 22e176e91812adea0b6cbd0b6630fea934f9dc47
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 24 23:27:46 2021 -0400

added fail tests

commit e8fbbbfeca4efe224cdf738c26284e978986f014
Author: Elisa Luo <eyl2130@columbia.edu>
Date: Sat Apr 24 20:06:09 2021 -0700

fixed codegen oops

commit ee633244541f5e8f673466dfdc8f92caded933a
Merge: 04b1d64 0a1d68c
Author: Elisa Luo <eyl2130@columbia.edu>
Date: Sat Apr 24 20:04:38 2021 -0700

merge

commit 04b1d64a74ec9f6c8565cf7aa1f75c473b4fbf97
Author: Elisa Luo <eyl2130@columbia.edu>
Date: Sat Apr 24 20:02:56 2021 -0700

started gpx functions

commit 0a1d68cbc74363155a8f0e6dd0dcefd8bcc33e5c
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 24 22:48:13 2021 -0400

fixed test-func

commit 0e81ffd01e83573d63653290f37d03ffcaa7d334
Merge: 7c80089 00cde18
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 24 22:36:14 2021 -0400

Merge branch 'main' of <https://github.com/ashleytkim/gwiz> into main

commit 7c8008985c6dea5abf0bf07890b77feb53d68c19
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 24 22:36:08 2021 -0400

added file test

commit 00cde1895193e0c2ae83d3688198e0f595415063
Author: Rebecca Yao <rebeccayao8@gmail.com>
Date: Sat Apr 24 22:34:36 2021 -0400

Adding function tests

commit ebe544f764e558eee17e427310432ef18e58e01f
Merge: a615963 cf4cae8
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 24 22:28:06 2021 -0400

added fail tests

commit a6159633afbce88801c954b9dc6f09c2951845c3
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 24 22:26:35 2021 -0400

added fail tests

commit cf4cae8258861448cb192bf564e0365e66de1aa1
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 24 20:52:26 2021 -0400

Update printout.sh

commit d6809873d576f2a8b665f52a8727912c18c8d1d5
Merge: 2441084 a67cecf
Author: Rebecca Yao <rebeccayao8@gmail.com>
Date: Sat Apr 24 20:37:54 2021 -0400

Added test cases to Makefile

commit 2441084a82667dfbc5bec319c7425e25d0608dff
Author: Rebecca Yao <rebeccayao8@gmail.com>
Date: Sat Apr 24 20:35:52 2021 -0400

Added test cases to Makefile

commit a67cecf4b5a838a5ae244341cdd24225bda20ace
Merge: 7952999 bf98072
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 24 18:03:55 2021 -0400

Merge branch 'main' of <https://github.com/ashleytkim/gwiz> into main

commit 7952999a70b67cd74efbc2b128365b074435300b
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 24 18:03:41 2021 -0400

resolved merge conflict

commit bf980724f557cc6753dcc11b50dbb26a2b64147e
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>

Date: Sat Apr 24 17:57:18 2021 -0400

Delete gwiz.native

commit did31a695b2622d4fbac01cea2d5e6e5bdb1bcad
Merge: be5dead e401faa
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 24 17:56:04 2021 -0400

Merge branch 'main' of <https://github.com/ashleytkim/gwiz> into main

commit be5deadceb73eb2c80c4799feb79725028ab754b
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 24 17:55:51 2021 -0400

updated shell script for fail-

commit e401faa528a427c7357ee6985441321c0f015804
Merge: 69aa0bf 5676b76
Author: Elisa Luo <eyl2130@columbia.edu>
Date: Sat Apr 24 12:35:56 2021 -0700

added built in funcs

commit 69aa0bf53a19f3560ba5ea066680fdc57c0b0301
Author: Elisa Luo <eyl2130@columbia.edu>
Date: Sat Apr 24 12:32:08 2021 -0700

added total distance and time functions

commit 5676b7669fca07b7c3442114c9ccc413cfdd2a70
Author: Rebecca Yao <rebeccayao8@gmail.com>
Date: Sat Apr 24 15:26:56 2021 -0400

Fixed error in codegen

commit 8f4a79f93d8417658f66134363638d55f30add08
Author: Rebecca Yao <rebeccayao8@gmail.com>
Date: Sat Apr 24 15:21:20 2021 -0400

Added file pointer type

commit f20b529ee36c9b7fe08a082c7fe34ba5136d5122
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 24 14:03:51 2021 -0400

Delete fail-assign1.err

commit 3c28ba761a4a65a019d1cf4ab613c87d095a7024
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>

Date: Sat Apr 24 14:02:41 2021 -0400

Delete gwiz.native

commit b226e4cd8520a084a7db3358854842e74e65b1e0
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 24 14:02:06 2021 -0400

Delete .DS_Store

commit 982170e26358e06f0a0f6a8837f1a928bdd725a5
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 24 14:01:45 2021 -0400

cleanup and resolved neg

commit 02ed4090aa4ea523127db8c84d466c8c58b430c4
Merge: 1fef016 611c3d9
Author: kpd2128 <kpd2128@barnard.edu>
Date: Sat Apr 24 07:30:31 2021 +0000

more TrackArray edits, still access error

commit 1fef016e3d03e584dd94f5c0728773a2ae04c3b7
Author: kpd2128 <kpd2128@barnard.edu>
Date: Sat Apr 24 07:29:50 2021 +0000

more TrackArray edits, still access error

commit 611c3d9404fd08d8b6028dd69fd37e05bb276232
Merge: c29c427 0bfabf0
Author: Rebecca Yao <rebccayao8@gmail.com>
Date: Fri Apr 23 23:43:53 2021 -0400

Merge branch 'main' of <https://github.com/ashleytkim/gwiz> into main

commit c29c4273283f13fcf40006a8fe8d4fb469b98670
Author: Rebecca Yao <rebccayao8@gmail.com>
Date: Fri Apr 23 23:42:00 2021 -0400

Adding shortened gpx file for testing

commit 0bfabf0d8a34460b0e0c399330cc7f4f7bd14720
Author: Elisa Luo <eyl2130@columbia.edu>
Date: Fri Apr 23 20:39:09 2021 -0700

update dockerfile

commit 127373286560c100136b0cf5ff6ac2965a8e22ac
Merge: 0a76010 95c7376

Author: Elisa Luo <eyl2130@columbia.edu>
Date: Fri Apr 23 20:29:01 2021 -0700

Merge branch 'main' of <https://github.com/ashleytkim/gwiz> into main

commit 0a760102363f83c25d09c423da0dad4101d7fa49
Author: Elisa Luo <eyl2130@columbia.edu>
Date: Fri Apr 23 20:28:11 2021 -0700

added linking to proper parse_gpx code, fixed issues w library and
makefile

commit 95c737614d9e08bc9d59a7dc9af75caa61678ede
Author: kpd2128 <kpd2128@barnard.edu>
Date: Sat Apr 24 01:25:44 2021 +0000

trackpoint test literal

commit a77cb42732cea039947a190cf8ff567ea21adca4
Author: kpd2128 <kpd2128@barnard.edu>
Date: Fri Apr 23 22:11:05 2021 +0000

test for track array

commit d8abf7ae8aaf323ca5060b5f9518f87623789238
Merge: 3ca484a 55b7852
Author: kpd2128 <kpd2128@barnard.edu>
Date: Fri Apr 23 22:03:02 2021 +0000

trackarrayliteral works in theory--assign error in semant for access

commit 3ca484a443b31c11fab19f002b11ef4f0a635e9b
Author: kpd2128 <kpd2128@barnard.edu>
Date: Fri Apr 23 22:02:17 2021 +0000

trackarrayliteral works in theory--assign error in semant for access

commit 55b785260716e0e2c792fcd07975ffc8eff0e5da
Author: Rebecca Yao <rebeccayao8@gmail.com>
Date: Fri Apr 23 16:44:54 2021 -0400

Added target parse_gpx in Makefile

commit e7bf4ba1953d332581ddea7226b8babcef24ab16
Author: Rebecca Yao <rebeccayao8@gmail.com>
Date: Fri Apr 23 14:01:19 2021 -0400

Added num_pt() function

commit 210dc12f181e92d01b0af51d27345595a2a24f09

Author: kpd2128 <kpd2128@barnard.edu>
Date: Fri Apr 23 08:59:30 2021 +0000

linked list array mostly complete. Seg fault on addEnd, fix tomorrow

commit 6b9ff8f5182095f86ef5c6783aee52a039bbac6b
Author: Elisa Luo <eyl2130@columbia.edu>
Date: Thu Apr 22 21:55:04 2021 -0700

tested file i/o for parsegpx

commit f0a61060cb326ba30160ea437e7a2959096be233
Author: Elisa Luo <eyl2130@columbia.edu>
Date: Tue Apr 20 16:11:34 2021 -0700

added linking to c files in shell script

commit 70d0b92ef0f1c48259c2f820e30f784f2585b591
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Apr 20 14:21:33 2021 -0400

Update README.md

commit 0ea352757193bdc4cf3456eea6396daf9e6253cf
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Apr 20 14:12:48 2021 -0400

Delete gwiz.native

commit 4c9f42442e7c45901b36cb34b4e5a881c2665050
Merge: 6265a5d fc284ed
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Apr 20 14:11:49 2021 -0400

added trackpoint

commit 6265a5d29919a0a99a7f030d33a209eb89fee1a7
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Apr 20 14:05:48 2021 -0400

added trackpoint

commit fc284edc582a370d11d2b705ef27ff7feb4287bc
Merge: 1c7119e 17aa786
Author: kpd2128 <kpd2128@barnard.edu>
Date: Tue Apr 20 00:14:47 2021 -0400

with array, still issues in codegen

commit 1c7119e6fab3e9584187755001a15503d21f8d89

Author: kpd2128 <kpd2128@barnard.edu>
Date: Tue Apr 20 00:04:12 2021 -0400

updated with array elements. Errors in codegen

commit 21578b01d58780ca6681277a1285bf432a2b45de
Author: kpd2128 <kpd2128@barnard.edu>
Date: Mon Apr 19 23:54:09 2021 -0400

c file and basic tests for array/lists

commit 17aa786b1a7404c5d5403103e533258af091e7fc
Author: Elisa Luo <eyl2130@columbia.edu>
Date: Sun Apr 18 21:31:58 2021 -0700

fixed return type of built in functions FINALLY

commit dce7c8e430de0e9f76d3fbc15c22e6462205ac10
Author: Rebecca Yao <rebeccayao8@gmail.com>
Date: Tue Apr 13 19:23:21 2021 -0400

Edited parse_gpx.c

commit 9d4cd43558a84a38ed201c34b1d10a3c5cd69c0c
Author: Rebecca Yao <rebeccayao8@Rebeccas-Air.lan>
Date: Tue Apr 13 19:02:34 2021 -0400

adding parse_gpx.c

commit 123ffd7fcbea1db21d61d0788196bc31f0d5ae1b
Author: Elisa Luo <eyl2130@columbia.edu>
Date: Sun Apr 11 17:00:41 2021 -0700

added parsegpx skeleton code and link to c function

commit be21623a5f89088b90f9329afa94008685da300b
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 10 15:59:06 2021 -0400

updated test with more loops

commit 9d11d45fe40384f5e4117e3c9b824d6b5a6ae5a9
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 10 15:58:44 2021 -0400

added trackpoint

commit 2e55b0804ef717ba4d59143cd98fe374cb21dd8a
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 10 15:58:38 2021 -0400

added trackpoint

commit 55f00a441dc125b559a1878fcbc7d23c3b5b8058
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 10 15:58:13 2021 -0400

added trackpoint

commit 137de9ed10965662331bb5554aeec85446d2481c
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 10 15:58:01 2021 -0400

added trackpoint

commit 9929f179257fd822656b104dd605514e8e1c791c
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 10 00:21:42 2021 -0400

deleted file

commit 47e5b86e2ec57cb2723b9c66d8da12ab4a4dd06f
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 10 00:21:15 2021 -0400

new extention

commit 6fc5f701072adc8cbfbf66f704d8464f852c43f2
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 10 00:21:01 2021 -0400

new test

commit 2f37aecb1383a001b0036e3b2df3ad0b58f0a156
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 10 00:20:42 2021 -0400

updated with .gw tests

commit d14d7c70c41576855898ef0a00c7c1000c8429bb
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 10 00:20:27 2021 -0400

works with assign, ifelse,rel ops

commit 9d6f47085ebf03c1a8746284c708d94ec7759722
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 10 00:20:21 2021 -0400

works with assign, ifelse,rel ops

commit f5561c4fb5b58a8a871129ec6ec772bd636d5879
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 10 00:20:12 2021 -0400

works with assign, ifelse,rel ops

commit 525663fec9890b2907ad29a08a765f6275d8637a
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 10 00:20:06 2021 -0400

updated with assign test

commit d01a4f464979e4b94695d47ecf28ea3e5d16c3f9
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 10 00:19:51 2021 -0400

works with assign, ifelse,rel ops

commit 44171910471642397de545915b4623291590b897
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 10 00:19:41 2021 -0400

works with assign, ifelse,rel ops

commit c4de06d239844af3d5ed8e936df26d639359a83e
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 10 00:19:31 2021 -0400

works with assign, ifelse,rel ops

commit 96c911fcc6a7e3f268d024de95b78ea43710a445
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 10 00:19:03 2021 -0400

works with assign, ifelse,rel ops

commit 4093a37ba1d97c7408f34ea5ff57c4672d0fce2
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Apr 10 00:18:04 2021 -0400

updated with .gw extention

commit 940d0c293865edb6b7c4ece7c55b65a109d082e7
Merge: 6123f58 5c92fbb
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Thu Apr 8 21:01:20 2021 -0400

updated makefile with fail

Merge branch 'main' of <https://github.com/ashleytkim/gwiz> into main

commit 6123f58b54ccde21e47990fab39a4a0ec2f3a8a9
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Thu Apr 8 21:01:02 2021 -0400

updated test

commit 5c92fbb60ff318392d54d15341e0089471cf8bae
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Thu Apr 8 21:00:07 2021 -0400

Delete helloworld directory

commit 7c99a7a9e6b4ed0424f8264383c6d98096a85951
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Thu Apr 8 20:59:57 2021 -0400

Delete testall.sh

commit ee1939ce1cdd591a2de53e699d782201f0ffe082
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Thu Apr 8 20:58:41 2021 -0400

added fail test section

commit b6746da15846f05666f5a034494f0c6aa2f6de56
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Thu Apr 8 20:56:33 2021 -0400

added fail test section

commit ee48a68b46fa8ab1e6a9274fe6729c7fb74ac3e5
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Thu Apr 8 20:54:05 2021 -0400

Delete test-helloworld.err

commit 200b33f403a556f20a908a33bc6a649d239ccd95
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Thu Apr 8 18:32:36 2021 -0400

Update README.md

commit 6f9b2c6a64e9e1861c056515d755fcf8b6d84917
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Thu Apr 8 18:30:14 2021 -0400

Update test-helloworld.out

commit 3e2638007757a743736063395379f7796095c051

Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Thu Apr 8 18:27:22 2021 -0400

Create test-helloworld.err

commit 114bb2564e50551c17fb46b5d1f37dbf6f86a3b8
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Thu Apr 8 18:26:58 2021 -0400

Create test-helloworld.out

commit 286085e52f94a471a78775d131046c13bde801df
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Thu Apr 8 18:26:35 2021 -0400

Rename test-helloworld.mc to test-helloworld.gz

commit 7a3ce9ccd30e98822039dbd26f1bcd4b18d75efb
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Thu Apr 8 18:23:01 2021 -0400

runs test folder and compares diff

commit e4bcc67e02532a3061500d96c951e949d8d8f8f6
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Thu Apr 8 18:21:11 2021 -0400

shell script compares diff

commit fc585adfd483b83591057e930bb784d5160ebb27
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Thu Apr 8 18:20:49 2021 -0400

shell script to print to terminal

commit e4e8e7ce868c27eb3fb763ad60cb04b5db7672dd
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Thu Apr 8 18:20:35 2021 -0400

make all, clean, test works

commit cf0228c307ab0f754b97df1e1627769cff3809ee
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Apr 6 23:49:10 2021 -0400

commented out for loop

commit bb8d893bf1ee7d3744628993eaa664be45a2efc1
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Apr 6 18:43:09 2021 -0400

new test for printing

commit a78f186c6d625cd8d513967efcf661875c646e60
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Apr 6 18:41:52 2021 -0400

supports print, added string and bool

commit 1f0b64e09a7bfdd318fe533cdd5b0069356b7b9f
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Apr 6 18:40:41 2021 -0400

modified to support print functions

commit bac4d4363c6824a004ea823ac2cf989e2a9dce12
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Apr 6 18:39:59 2021 -0400

modified to support print functions

commit 77b7be60866ced65488090a5dbc16b0b7b2ee3b9
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Apr 6 18:39:05 2021 -0400

modified to support print functions

commit ef921014ddbc24b3bcc00bcac1d86cc6459fba73
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Apr 6 18:37:59 2021 -0400

supports print, added operators

commit 8c49a399fb03864e6e712bd89f62de09d11ee36c
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Apr 6 18:36:21 2021 -0400

modified to support print functions

commit 7fb1e7f4082b5625692170880f8bc3399eca2254
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Apr 6 18:35:16 2021 -0400

Update ast.ml

commit 9eebd1fe1e268962b2dfc8e373ab8af28879ceb9
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Apr 6 17:58:07 2021 -0400

Rename helloworld/tests/test-helloworld.mc to

tests/test-helloworld.mc

commit 628ebb664c033c715664c7e92710bca413db6ad7
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Apr 6 17:57:39 2021 -0400

Rename helloworld/testhello.sh to testhello.sh

commit c5fa6dc4a45779d437a72130cb60c6502853ddb1
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Mon Apr 5 23:31:03 2021 -0400

Rename helloworld/Makefile to Makefile

commit 47160981ff22ed5d16f79003e0397a5d46e4064e
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Mon Apr 5 23:25:07 2021 -0400

Rename helloworld/semant.ml to semant.ml

commit 668b09523de92d6e36e46737f0a9ac87d37fe62f
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Mon Apr 5 23:24:54 2021 -0400

Rename helloworld/scanner.mll to scanner.mll

commit ef62f0d32df8d7210f21f033178cc3e47da62ace
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Mon Apr 5 23:24:40 2021 -0400

Rename helloworld/parser.mly to parser.mly

commit 9482f0de073c2f3661625d20ecca3afe4683e5ea
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Mon Apr 5 23:24:25 2021 -0400

Rename helloworld/gwiz.ml to gwiz.ml

commit b6132a177aeb16f96792f6f1a78d6a890d62438e
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Mon Apr 5 23:24:09 2021 -0400

Rename helloworld/sast.ml to sast.ml

commit ea0b4e02f8d3ef9624542c3754b704cd6459f38a
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Mon Apr 5 23:23:50 2021 -0400

Rename helloworld/codegen.ml to codegen.ml

commit 048a557e55dd0ce82569f102f9fc88faee4a6dff
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Mon Apr 5 23:23:36 2021 -0400

Rename helloworld/ast.ml to ast.ml

commit ac56744645eb6a113d7dff30806bd55955a56c2c
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Mon Apr 5 23:23:23 2021 -0400

Rename helloworld/_tags to _tags

commit 47d69ffc0a6d6037ebfa1f7a9411984aadb5643f
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Mon Apr 5 23:22:53 2021 -0400

Rename helloworld/Gwizparse.mly to gwizparse.mly

commit 043de0ae26058197646e656c17ad32548311fbf7
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Mon Apr 5 23:22:15 2021 -0400

Rename helloworld/Dockerfile to Dockerfile

commit 6eeb69af1682e79cc4f5878a9851f674575b862e
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Mon Apr 5 23:21:33 2021 -0400

Delete codegen.ml

commit c6a3f6fb16e89a78c39bfd8f4b58833331dcb85
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Mon Apr 5 23:21:26 2021 -0400

Delete parser2.mly

commit d86d0d73f6cc8371dc7f89936a107f8995e7ceac
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Mon Apr 5 23:21:16 2021 -0400

Delete gwiz.ml

commit ff0b8e70c929eeef7b4a2e1f36d4bf7841fad9de
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Mon Apr 5 23:20:51 2021 -0400

Delete parser.mly

commit 818ef692ed59fcc9359ce5f45917c2a90aff8af8
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>

Date: Mon Apr 5 23:20:44 2021 -0400

Delete scanner.mll

commit 4635f2dc200abcea02e950303467e1fc9308388b
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Mon Apr 5 23:20:33 2021 -0400

Delete ast.ml

commit 051cd4f72b4d2cfba97724e4bd6e4faed478102b
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Mon Apr 5 23:20:15 2021 -0400

Delete ast2.ml

commit 723363049c04c99ec1e1fd8a6869f01b34fc4ba7
Author: Rebecca Yao <rebeccayao8@Rebeccas-Air.lan>
Date: Sat Apr 3 20:43:56 2021 -0400

Edited so that assign and double now work

commit d9bab6d322eaae4db30349dbe80690ea9db13812
Author: kpd2128 <katieduff02@DESKTOP-61K68RC.localdomain>
Date: Tue Mar 30 15:59:22 2021 -0400

assign attempt #1

commit 3007fd1797ab9ac7902aabe2c34a2bba25b8adf4
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Wed Mar 24 23:19:36 2021 -0400

Update README.md

commit a962f9dbe5d892c86828a78a5f69d3a1cf8ee618
Merge: fde1b39 1f57621
Author: Rebecca Yao <rebeccayao8@Rebeccas-Air.lan>
Date: Wed Mar 24 19:50:58 2021 -0400

Merge branch 'main' of <https://github.com/ashleytkim/gwiz> into main

commit 1f57621ad06506e4905d9a5c1d4eaba8007d7c4f
Author: kpd2128 <43865750+kpd2128@users.noreply.github.com>
Date: Wed Mar 24 19:50:35 2021 -0400

Update README.md

commit fde1b3961d38889fbdbe8add79f96fff40ca3d58
Author: Rebecca Yao <rebeccayao8@Rebeccas-Air.lan>
Date: Wed Mar 24 19:50:34 2021 -0400

Edited test-helloworld.mc and testhello.sh

commit c25b7b903bcf7cc3670816cd6a199d35752ffb75
Author: kpd2128 <43865750+kpd2128@users.noreply.github.com>
Date: Wed Mar 24 19:36:28 2021 -0400

Update README.md

commit 34c330109ac9584e5fa023e5205d8435079409c4
Author: kpd2128 <43865750+kpd2128@users.noreply.github.com>
Date: Wed Mar 24 17:18:45 2021 -0400

Update testhello.sh

commit ea7b5349764b4dd7e0a550abcb0556f5aeac73d1
Author: Elisa Luo <64823602+valgrind1@users.noreply.github.com>
Date: Wed Mar 24 14:04:03 2021 -0700

test hello

commit eb01e53836e6c76df8ac66def3d20af7bd6ec905
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Wed Mar 24 16:44:40 2021 -0400

Update Makefile

commit 89a983e79d452f3d2121b792053f47198ce5028f
Author: Elisa Luo <64823602+valgrind1@users.noreply.github.com>
Date: Wed Mar 24 13:41:57 2021 -0700

hello world works !!11

commit fd885ea2d2c8f4b2d28d50b3c428dd63cff67814
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Wed Mar 24 15:27:17 2021 -0400

Update testall.sh

commit b9482f55258d99cfe1f19b760d74e9c85b1ed272
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Wed Mar 24 15:22:41 2021 -0400

Delete test.sh

commit 1fd17813445217b2bd849f5b448762a3eb6df9f2
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Wed Mar 24 15:22:02 2021 -0400

Update Makefile

commit e213e782f5ec4cc2f9c0516f8537b5f40f9a9309
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Wed Mar 24 15:20:56 2021 -0400

Update Makefile

commit 2a0a4d5c2fe6b9b1d4a11d31763fe22cf43505b3
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Wed Mar 24 15:20:37 2021 -0400

Create test-helloworld.mc

commit 36d0a3f7f03147f242f7f2583a27faed94bbd114
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Wed Mar 24 15:19:02 2021 -0400

Update Makefile

commit 10f7aa37b03c7391627ffbc03d5e9aede7aa2ede
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Wed Mar 24 15:18:14 2021 -0400

Update semant.ml

commit bda914ae9f132e04fae0ab2b3ee8b905205ca41d
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Wed Mar 24 15:16:57 2021 -0400

Delete testhello.sh

commit c4b498fd03241270a2089c569749951fb3c3708f
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Wed Mar 24 15:16:47 2021 -0400

Rename testall to testall.sh

commit b0f21b6fd6e6f8e411549184c6c761ac6428fa06
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Wed Mar 24 15:16:33 2021 -0400

Create testall

commit ee1ce26e66de10b8cd63b6a6ea94c8cddd672f2e
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Wed Mar 24 13:08:54 2021 -0400

Update semant.ml

commit be94eac5c967fc01a11cba98d88fd6d341a910a2

Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Wed Mar 24 11:04:24 2021 -0400

Add files via upload

commit 2e4237489ac3db064959f13374a364e23b744bff
Author: Elisa Luo <64823602+valgrind1@users.noreply.github.com>
Date: Tue Mar 23 21:17:15 2021 -0700

helloworld

commit 6ee9bfa59655c6ddc35e836cf42f6fd90818aee3
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Mar 23 23:35:42 2021 -0400

Update and rename sast.ml to helloworld/sast.ml

commit 00b1c8b5de6a8715814039efd8e9231cd8d3447a
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Mar 23 23:34:19 2021 -0400

Delete sast.ml

commit 3b8ed64be1b09cefa01e0326974bd2fdfa81e6bc
Author: Elisa Luo <64823602+valgrind1@users.noreply.github.com>
Date: Tue Mar 23 20:32:52 2021 -0700

Add files via upload

you wish that it compiles

commit 5d160076a63e716cd81de8e9f3e98c2fc8b59e1f
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Mar 23 23:29:36 2021 -0400

updated string_of_sfdecl with argument fdecl

commit 68cfa9126d568a882778af9eb4bc9103edc2a0e7
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Mar 23 23:27:45 2021 -0400

Update sast.ml

commit f746a435e403b6a64ca168834b531cad76f66435
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Mar 23 23:25:50 2021 -0400

changed all stmt to sstmt

commit 77a37a11d84d7447b7207730e7b86bb9795097b5

Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Mar 23 23:22:09 2021 -0400

line 16 edit to sstmt

commit ea4420c01d1409d80b71a69adade491283edb877
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Mar 23 23:17:20 2021 -0400

line 60 renamed function

commit 3626edd689828e0a313adf9a05064d7495b20906
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Mar 23 23:13:07 2021 -0400

line 67 changed to string_of_sstmt

commit 9166a50a54b22463652006c537096d6b861873e5
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Mar 23 23:10:11 2021 -0400

added string_of_vdecl line 60

commit eaa2d5b36b3c9a894a84a86e967650d97cae316e
Author: Rebecca Yao <rebeccayao8@Rebeccas-Air.lan>
Date: Tue Mar 23 21:01:49 2021 -0400

Adding testhello.sh

commit f012baddeef935732c0b149bfafc23da5a83e71f
Merge: edcef4a f68b905
Author: Rebecca Yao <rebeccayao8@Rebeccas-Air.lan>
Date: Tue Mar 23 18:37:59 2021 -0400

Merge branch 'main' of <https://github.com/ashleytkim/gwiz> into main

commit edcef4ae2a5bc2933da38516063ad454392a1ae5
Merge: ecd320f 6f2242a
Author: Rebecca Yao <rebeccayao8@Rebeccas-Air.lan>
Date: Tue Mar 23 18:37:49 2021 -0400

Merge branch 'shell-testing' into main

commit 6f2242a4f33fdcd14ae9eebca5e8eafab6ec696e
Author: Rebecca Yao <rebeccayao8@Rebeccas-Air.lan>
Date: Tue Mar 23 18:37:38 2021 -0400

Adding test.sh

commit f68b9051887caffb9a374af56191c99fe76cce28

Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Mar 23 18:29:08 2021 -0400

Update with comments on modifications

commit e4ba09583e9abc571cc7f4f875c0ae4a7ccfb7d7
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Mar 23 18:27:15 2021 -0400

Rename readme.md to README.md

commit 02c90ce24a92929a9b93157449bd90c4256a55ae
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Mar 23 18:26:52 2021 -0400

Rename hwscanner.mll to scanner.mll

commit 2512a7f6d5e9085c2b73e28544c7248c5d3f2a10
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Mar 23 18:26:30 2021 -0400

Rename hwparser.mly to parser.mly

commit 797204b0aec1e84148ba8aad00379f03d51bd9d
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Mar 23 18:26:13 2021 -0400

Delete hwast.ml

commit 58107d1540449c364c8f8a25a73e1a61e3983eec
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Mar 23 18:25:59 2021 -0400

Rename sast.ml to helloworld/sast.ml

commit 0925eee9a131f10b452dac5616b5508f6e167333
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Mar 23 18:25:43 2021 -0400

removed double, char, updated assign to match AST

commit ecd320f129614e8fa53b619d42df590d4800f275
Merge: 38c211b 688d1ef
Author: Rebecca Yao <rebeccayao8@Rebeccas-Air.lan>
Date: Mon Mar 22 23:58:43 2021 -0400

Merge branch 'main' of <https://github.com/ashleytkim/gwiz> into main

commit 38c211b53686015326900160a19b3cd30b781a82
Author: Rebecca Yao <rebeccayao8@Rebeccas-Air.lan>

Date: Mon Mar 22 23:55:30 2021 -0400

Edited Makefile and renamed ast.ml

commit 688d1ef766477f19c65ba55d0d6e984422269a89
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Mon Mar 22 23:55:12 2021 -0400

Update hwparser.mly

Added a missing |

commit 4e4a226ee84464ba5e83f913800560272ca9b8a6
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Mon Mar 22 20:09:17 2021 -0400

Create Makefile

commit 0352ce4de9966cf19863e2bd0fc1112545f3e6f7
Author: kpd2128 <43865750+kpd2128@users.noreply.github.com>
Date: Mon Mar 22 20:02:37 2021 -0400

Update hwparser.mly

commit 5ff97e368e57f9e9bf1c60055b7501b0041bc772
Author: kpd2128 <43865750+kpd2128@users.noreply.github.com>
Date: Mon Mar 22 14:35:53 2021 -0400

Update hwparser.mly

commit 3ee42087bae90a8d232f5cf8c1c028e1c9a7edca
Author: kpd2128 <43865750+kpd2128@users.noreply.github.com>
Date: Mon Mar 22 14:34:15 2021 -0400

Update hwast.ml

commit 3afe5780b4c7d0932b64dc91fee974e77e2570c6
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Sun Mar 21 18:41:31 2021 -0400

Delete .DS_Store

commit 85a8363eb3687b4e447124d4e4c1c0ce3d7153cb
Author: kpd2128 <43865750+kpd2128@users.noreply.github.com>
Date: Sun Mar 21 18:38:38 2021 -0400

Update hwscanner.mll

commit 2d275b4e363fe85c8734477ac12e7dae9aab98a2
Author: kpd2128 <43865750+kpd2128@users.noreply.github.com>

Date: Sun Mar 21 18:37:49 2021 -0400

Create hwscanner.mll

commit 875a7ba8770d15744bf92a1fab75436527ff5db7
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Sun Mar 21 18:34:58 2021 -0400

Update readme.md

commit 0d1f845b0ac347d8c7c206602fb3e28076c392f3
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Sun Mar 21 18:34:39 2021 -0400

Rename parser.mly to hwparser.mly

commit 326da8a2dadff545f0539778e2068040f4d4478e
Author: kpd2128 <43865750+kpd2128@users.noreply.github.com>
Date: Sun Mar 21 18:34:26 2021 -0400

Update and rename ast3.ml to hwast.ml

commit 8873839550813d3fdcca229ee96c878e1bace007
Author: kpd2128 <43865750+kpd2128@users.noreply.github.com>
Date: Sun Mar 21 18:33:53 2021 -0400

Update and rename parser3.mly to parser.mly

commit f3afd643c70097bf1419dc286727c9fcc7ae2f09
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Sun Mar 21 18:25:47 2021 -0400

Rename parser3.mly to helloworld/parser3.mly

commit 1111dd187ce0ea35cb58440b5929a0f9e18b596d
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Sun Mar 21 18:25:29 2021 -0400

Rename ast3.ml to helloworld/ast3.ml

commit 040c532885ae2a8cda6a605fc6afa8df99497484
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Sun Mar 21 18:24:02 2021 -0400

Create readme.md

commit 9efa6ecc69e4e96388b0c54cd32e9e010bb3c706
Author: kpd2128 <43865750+kpd2128@users.noreply.github.com>
Date: Sun Mar 21 18:21:21 2021 -0400

Create ast3.ml

commit 26644a42a7fd5e758ffdb8401be4c7ec139d583c
Author: kpd2128 <43865750+kpd2128@users.noreply.github.com>
Date: Sun Mar 21 18:20:59 2021 -0400

Create parser3.mly

commit 1313e077cf8a8c527bb0942168cf0fa4129677b7
Author: kpd2128 <43865750+kpd2128@users.noreply.github.com>
Date: Sun Mar 21 17:26:45 2021 -0400

Create ast2.ml

commit 784465597e64ded07372edca7d67418fbe67adc4
Author: kpd2128 <43865750+kpd2128@users.noreply.github.com>
Date: Sun Mar 21 17:26:19 2021 -0400

Create parser2.mly

commit 06a82fa6217061097a5d140e253da0892ad13505
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Mar 20 17:02:54 2021 -0400

Update README.md

commit fb89956f61b8f005fb85e013f200bf92cdc0a9cc
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Mar 20 17:00:38 2021 -0400

Update README.md

commit f6ef7131e3e98cb960493cc96fb831c6d8aa6d13
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Mar 20 16:59:54 2021 -0400

Update README.md

commit d74776f4c523e8863640d2ba75ee06ec1712132e
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Mar 20 16:59:37 2021 -0400

Update README.md

commit 911f780dfdf9ec6b13a99669eff8143190e4032a
Author: Ashley Kim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Mar 20 16:59:21 2021 -0400

Create README.md

commit 7998c2d188cdebb405941ec650c33f4e5cff8104
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Mar 20 16:56:27 2021 -0400

3/20 first commit

commit 75f99378a44108adca02878e92057317d3c72f4f
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Sat Mar 20 16:55:47 2021 -0400

left comment for potential let rec string_of_expr expr =

commit 64811c27fc30fdb1462e2daaf29411ca7a60de33
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Thu Mar 18 20:00:25 2021 -0400

first ast commit

commit ece9491fca3081559602a44264f5e00ec7644c0e
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Mar 9 17:50:34 2021 -0500

aligned comments

commit 9aaa82e51141c7feff7d7f104eb14f2566734b94
Author: ashleytkim <57121181+ashleytkim@users.noreply.github.com>
Date: Tue Mar 9 17:43:55 2021 -0500

3/9 first commit

10.2 gwiz.ml

Author: Elisa

(Top-level of the GWiz compiler: scan & parse the input,
check the resulting AST and generate an SAST from it, generate LLVM IR,
and dump the module
Based on MicroC
File: gwiz.ml
Author: Elisa Luo
PLT Spring 2021 *)*

type action = Ast | Sast | LLVM_IR | Compile

```
let () =  
  let action = ref Compile in  
  let set_action a () = action := a in  
  let speclist = [
```

```

    ("-a", Arg.Unit (set_action Ast), "Print the AST");
    ("-s", Arg.Unit (set_action Sast), "Print the SAST");
    ("-l", Arg.Unit (set_action LLVM_IR), "Print the generated LLVM IR");
    ("-c", Arg.Unit (set_action Compile),
     "Check and print the generated LLVM IR (default)");
  ] in
let usage_msg = "usage: ./microc.native [-a/-s/-l/-c] [file.mc]" in
let channel = ref stdin in
Arg.parse speclist (fun filename -> channel := open_in filename) usage_msg;

let lexbuf = Lexing.from_channel !channel in
let ast = Gwizparse.program Scanner.token lexbuf in
match !action with
  Ast -> print_string (Ast.string_of_program ast)
| _ -> let sast = Semant.check ast in
      match !action with
        Ast -> ()
      | Sast -> print_string (Sast.string_of_sprogram sast)
      | LLVM_IR -> print_string (Llvm.string_of_llmodule (Codegen.translate sast))
      | Compile -> let m = Codegen.translate sast in
                  Llvm_analysis.assert_valid_module m;
                  print_string (Llvm.string_of_llmodule m)

```

10.3 scanner.mll

Authors: Ashley, Katherine

```

(*)
  OCamlLex scanner for GWiz
  File: scanner.mll
  Authors: Ashley Kim, Katherine Duff
  PLT Spring 2021

*)

{ open Gwizparse }

(* Definitions *)
let digit = ['0' - '9']
let letter = ['a' - 'z' 'A' - 'Z']
let digits = digit+
let simple_char = [' ' - '! ' # - '€' ' ( - '[' ' ] - '~ ' ]
let escape_char = ['t' 'r' 'n' '\\ ' '\"' '\\\']

(* Rules *)
rule token = parse

```

```

    [' ' '\t' '\r'] {token lexbuf}
| "/*" { comment lexbuf }
| '\n' { Lexing.new_line lexbuf; token lexbuf }

(* Blocks, Delimiters, Terminators *)
| '(' {LPAREN}
| ')' {RPAREN}
| '{' {LBRACE}
| '}' {RBRACE}
| '[' {LBRAC}
| ']' {RBRAC}
| ';' {SEMI}
| '.' {PERIOD}
| ',' {COMMA}

(* Arithmetic Operators *)
| '+' {PLUS}
| '-' {MINUS}
| '*' {TIMES}
| '/' {DIVIDE}

(* Assignment Operators *)
| '=' {ASSIGN}

(* Relational Operators *)
| "==" {EQ}
| "!=" {NEQ}
| "<" {LT}
| "<=" {LEQ}
| ">" {GT}
| ">=" {GEQ}

(* Logical Operators *)
| "and" {AND}
| "or" {OR}
| "not" {NOT}
| "-" {NEG}

(* Keywords *)
| "if" {IF}
| "else" {ELSE}
| "for" {FOR}
| "while" {WHILE}
| "return" {RETURN}
| "int" {INT}
| "double" {DOUBLE}

```

```

| "bool"          {BOOL}
| "string"       {STRING}
| "void"         {VOID}
| "Trackpoint"  {TRACKPOINT}
| "file"         {FILE}

(* Literals *)
| "true"          {BLIT(true)}
| "false"         {BLIT(false)}
| digits as lit   {LITERAL(int_of_string lit)}
| digits '.' digit* ( ['e' 'E'] ['+' '-']? digits )? as lit {DLIT(lit)}
| letter (letter | digit)* as lit {ID(lit)}
| '\"' ((simple_char | '\\ ' escape_char)* as lit) '\"' {SLIT(lit)}
| eof            {EOF}

and comment = parse
    "*/"          { token lexbuf }
| _              { comment lexbuf }

```

10.4 gwizparse.mly

Authors: Ashley, Elisa, Katherine

```

/* Ocaml yacc parser for GWiz
   Based on MicroC
   File: gwizparse.mly
   Authors: Ashley Kim, Elisa Luo, Katherine Duff
   PLT Spring 2021
*/

%{
  open Ast
%}

%token SEMI LPAREN RPAREN LBRACE RBRACE LBRAC RBRAC PERIOD COMMA /*seperators*/
%token PLUS MINUS TIMES DIVIDE ASSIGN /*binary operators */
%token NOT EQ NEQ LT LEQ GT GEQ AND OR NEG /*logical operators*/
%token IF ELSE FOR WHILE RETURN INT BOOL DOUBLE STRING VOID TRACKPOINT FILE /*keywords*/
%token <int> LITERAL
%token <bool> BLIT /*boolean literal*/
%token <string> ID SLIT DLIT
%token EOF

%nonassoc NOELSE
%nonassoc ELSE

```



```

%left SEMI
%right ASSIGN
%left PERIOD
%left OR
%left AND
%left EQ NEQ
%left LT GT LEQ GEQ
%left PLUS MINUS
%left TIMES DIVIDE
%right NOT
%right NEG

%start program
%type <Ast.program> program

%%

program:
  decls EOF { $1 }

decls:
  /*nothing*/ { ([], []) }
  | decls vdecl { (($2 :: fst $1), snd $1) }
  | decls fdecl { (fst $1, ($2 :: snd $1)) }

fdecl: /* function declaration */
  typ ID LPAREN formals_opt RPAREN LBRACE vdecl_list stmt_list RBRACE
  { { typ = $1;
    fname = $2;
    formals = List.rev $4;
    locals = List.rev $7;
    body = List.rev $8 } }

formals_opt:
  /* nothing */ { [] }
  | formal_list { $1 }

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

typ:
  INT { Int }
  | DOUBLE { Double }

```

```

| BOOL { Bool }
| VOID { Void }
| STRING { String }
| TRACKPOINT { Trackpoint }
| FILE { File }
| typ LBRAC RBRAC { ArrayType($1) }

vdecl_list:
    /*nothing*/ { [] }
| vdecl_list vdecl { $2 :: $1 }

vdecl: /*variable declaration */
    typ ID SEMI { ($1, $2) }

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

stmt: /*statements*/
    expr SEMI { Expr ($1) }
| RETURN expr_opt SEMI { Return ($2)}
| LBRACE stmt_list RBRACE { Block(List.rev $2)} /*nesting statements*/
| 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)          }

expr_opt: /*when an expression is optional*/
    /*nothing*/ { Noexpr }
| expr { $1 }

expr:
    LITERAL { Literal($1) }
| DLIT { DLit($1) }
| BLIT { BLit($1) }
| ID { Id($1) }
| SLIT { SLit($1) }
| expr PLUS expr { Binop($1, Add, $3) }
| 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 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) }
| ID LPAREN args_opt RPAREN { Call($1, $3) } /*function calling*/
| LPAREN expr COMMA expr COMMA expr RPAREN { TrackpointLit($2, $4, $6) }
| expr PERIOD ID { TrackpointAccess($1, $3) }
| LBRAC args_opt RBRAC { ArrayLiteral($2) }
| expr MINUS LT expr GT { ArrayAccess($1, $4)}
| expr PLUS LT expr GT { ArrayAppend($1, $4) }

```

```

args_opt:
  /*nothing*/ { [] }
  | args_list { List.rev $1 }

```

```

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

```

10.5 ast.ml

Authors: Ashley, Katherine

```

(*
  Abstract Syntax Tree for Gwiz
  Based on MicroC
  File: ast.ml
  Authors: Ashley Kim, Katherine Duff
  PLT Spring 2021
*)

```

```

type op = Add | Sub | Mult | Div | Equal | Neq | Less | Leq | Greater | Geq | And | Or

```

```

type typ = Int | String | Bool | Void | Double | Trackpoint | ArrayType of typ (*/ TrackArrr
type bind = typ * string

```

```

type uop = Neg | Not

```

```

type expr =
  Literal of int
  | DLit of string
  | SLit of string

```

```

| BLit of bool
| Id of string
| Binop of expr * op * expr
| Unop of uop * expr
| Assign of string * expr
| Call of string * expr list
| TrackpointLit of expr * expr * expr
| TrackpointAccess of expr * string
| ArrayLiteral of expr list
| ArrayAccess of expr * expr
| ArrayAppend of expr * expr
(*
 / TrackArrayLiteral of expr list
 / TrackArrayAccess of expr * expr
 / TrackArrayInit of expr * int *)
| Noexpr

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

type func_decl = {
  typ : typ;
  fname : string;
  formals : bind list;
  locals : bind list;
  body : stmt list;
}

type program = bind list * func_decl list

(* Pretty-printing functions. *)
let string_of_op = function
  Add -> "+"
  | Sub -> "-"
  | Mult -> "*"
  | Div -> "/"
  | Equal -> "=="
  | Neq -> "!="
  | Less -> "<"

```

```

| Leq -> "<="
| Greater -> ">"
| Geq -> ">="
| And -> "∧"
| Or -> "∨"

let string_of_uop = function
  Neg -> "-"
  | Not -> "!"

let rec string_of_expr = function
  Literal(l) -> string_of_int l
  | DLit(l) -> l
  | BLit(true) -> "true"
  | BLit(false) -> "false"
  | SLit(l) -> l
  | 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) ^ ")"
  | TrackpointAccess(e1, s2) -> string_of_expr e1 ^ "." ^ s2
  | TrackpointLit(e1, e2, e3) -> "(" ^ string_of_expr e1 ^ ",
" ^ string_of_expr e2 ^ ", " ^ string_of_expr e3 ^ ")"
  | ArrayLiteral(e1) -> "[" ^ String.concat ", " (List.map string_of_expr e1) ^ "]"
  | ArrayAccess(v, e) -> string_of_expr v ^ "-<" ^ string_of_expr e ^ ">"
  | ArrayAppend(v, e) -> string_of_expr v ^ "+<" ^ string_of_expr e ^ ">"
  (*| TrackArrayLiteral (e1) -> "<[" ^ String.concat ",
" (List.map string_of_expr e1) ^ "]">"
  | TrackArrayAccess(v,e) -> string_of_expr v ^ "<[" ^ string_of_expr e ^ "]">"
  | TrackArrayInit(i, s) -> "new" ^ string_of_expr i ^ "<[" ^ string_of_int s ^ "]">" *)
  | Noexpr -> ""

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

let rec string_of_typ = function
  Int -> "int"
| Double -> "double"
| Bool -> "bool"
| String -> "string"
| Void -> "void"
| Trackpoint -> "Trackpoint"
| ArrayType(t) -> string_of_typ t ^ "[]"
(* | TrackArrayType(t, l) -> string_of_typ t ^ "<[" ^ string_of_int l ^ "]">" *)
| File -> "file"

let string_of_vdecl (t, id) = string_of_typ t ^ " " ^ id ^ "; \n"

let string_of_fdecl fdecl =
  string_of_typ fdecl.typ ^ " " ^
  fdecl.fname ^ "(" ^ String.concat ", " (List.map snd fdecl.formals) ^
  ") \n{ \n" ^
  String.concat "" (List.map string_of_vdecl fdecl.locals) ^
  String.concat "" (List.map string_of_stmt fdecl.body) ^
  "} \n"

let string_of_program (vars, funcs) =
  String.concat "" (List.map string_of_vdecl vars) ^ " \n" ^
  String.concat " \n" (List.map string_of_fdecl funcs)

```

10.6 sast.ml

Authors: Ashley, Katherine

```

(*)
  Semantically checked Abstract Syntax Tree for Guiz
  Based on MicroC
  File: sast.ml
  Authors: Ashley Kim, Katherine Duff
  PLT Spring 2021
*)

```

```
open Ast
```

```

type sexpr = typ * sx
and sx =
  SLiteral of int

```

```

| SDLit of string
| SSLit of string
| SBLit of bool
| SId of string
| SBinop of sexpr * op * sexpr
| SUnop of uop * sexpr
| SAssign of string * sexpr
| SCall of string * sexpr list
| STrackpointLit of sexpr * sexpr * sexpr
| STrackpointAccess of sexpr * string
| SArrayLiteral of sexpr list
| SArrayAccess of sexpr * sexpr
| SArrayAppend of sexpr * sexpr
(*| STrackArrayLiteral of sexpr list
 / STrackArrayAccess of sexpr * sexpr
 / STrackArrayInit of sexpr * sexpr *)
| SNoexpr

type sstmt =
  SBlock of sstmt list
  | SExpr of sexpr
  | SReturn of sexpr
  | SIf of sexpr * sstmt * sstmt
  | SFor of sexpr * sexpr * sexpr * sstmt
  | SWhile of sexpr * sstmt

type sfunc_decl = {
  styp : typ;
  sfname : string;
  sformals : bind list;
  slocals: bind list;
  sbody : sstmt list;
}

type sprogram = bind list * sfunc_decl list

(* Pretty-printing functions *)

let rec string_of_sexpr (t,e) =
  "(" ^ string_of_typ t ^ " : " ^ (match e with
    SLiteral(l) -> string_of_int l
  | SDLit(l) -> l
  | SBLit(true) -> "true"
  | SBLit(false) -> "false"
  | SSLit(l) -> l
  | SId(s) -> s

```

```

| SBinop(e1, o, e2) -> string_of_sexpr e1 ^ " " ^ string_of_op o ^ "
" ^ string_of_sexpr e2
| SUnop(o, e) -> string_of_uop o ^ string_of_sexpr e
| SAssign(v, e) -> v ^ " = " ^ string_of_sexpr e
| SCall(f, el) ->
  f ^ "(" ^ String.concat ", " (List.map string_of_sexpr el) ^ ")"
| STrackpointAccess(s1, s2) -> string_of_sexpr s1 ^ "." ^ s2
| STrackpointLit(e1, e2, e3) -> "(" ^ string_of_sexpr e1 ^ ",
" ^ string_of_sexpr e2 ^ ", " ^ string_of_sexpr e3 ^ ")"
| SArrayLiteral(e1) -> "[" ^ String.concat ", " (List.map string_of_sexpr e1) ^ "]"
| SArrayAccess(v, e) -> string_of_sexpr v ^ "-<" ^ string_of_sexpr e ^ ">"
| SArrayAppend(v, e) -> string_of_sexpr v ^ "+<" ^ string_of_sexpr e ^ ">"
(*| STrackArrayLiteral(e1) -> "<[" ^ String.concat ",
" (List.map string_of_sexpr e1) ^ ">"
| STrackArrayAccess(v, e) -> string_of_sexpr v ^ "<[" ^ string_of_sexpr e ^ ">"
| STrackArrayInit(i, s) -> "new" ^ string_of_sexpr i ^ "<[" ^ string_of_sexpr s ^ ">" *)
| SNoexpr -> ""
                                ) ^ ")"

let rec string_of_sstmt = function
  SBlock(stmts) ->
    "{\n" ^ String.concat "" (List.map string_of_sstmt stmts) ^ "}\n"
  | SExpr(expr) -> string_of_sexpr expr ^ ";\n";
  | SReturn(expr) -> "return " ^ string_of_sexpr expr ^ ";\n";
  | SIf(e, s, SBlock([])) -> "if (" ^ string_of_sexpr e ^ ")\n" ^ string_of_sstmt s
  | SIf(e, s1, s2) -> "if (" ^ string_of_sexpr e ^ ")\n" ^
    string_of_sstmt s1 ^ "else\n" ^ string_of_sstmt s2
  | SFor(e1, e2, e3, s) ->
    "for (" ^ string_of_sexpr e1 ^ " ; " ^ string_of_sexpr e2 ^ " ; " ^
    string_of_sexpr e3 ^ ") " ^ string_of_sstmt s
  | SWhile(e, s) -> "while (" ^ string_of_sexpr e ^ ") " ^ string_of_sstmt s

let string_of_sfdecl fdecl =
  string_of_typ fdecl.styp ^ " " ^
  fdecl.sfname ^ "(" ^ String.concat ", " (List.map snd fdecl.sformals) ^
  ")\n{\n" ^
  String.concat "" (List.map string_of_vdecl fdecl.slocals) ^
  String.concat "" (List.map string_of_sstmt fdecl.sbody) ^
  "}\n"

let string_of_sprogram (vars, funcs) =
  String.concat "" (List.map string_of_vdecl vars) ^ "\n" ^
  String.concat "\n" (List.map string_of_sfdecl funcs)

```


10.7 semant.ml

Authors: Ashley, Katherine

```
(*
  Semantic checking for the GWiz compiler
  Based on MicroC
  File: semant.ml
  Authors: Ashley Kim, Katherine Duff
  PLT Spring 2021
*)

open Ast
open Sast

module StringMap = Map.Make(String)

(* Semantic checking of the AST. Returns an SAST if successful,
   throws an exception if something is wrong.

   Check each global variable, then check each function *)

let check (globals, functions) =

  (* 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))
      | _ -> ()) binds;
    let rec dups = function
      [] -> ()
      | ((_,n1) :: (_,n2) :: _) when n1 = n2 ->
          raise (Failure ("duplicate " ^ kind ^ " " ^ n1))
      | _ :: t -> dups t
    in dups (List.sort (fun (_,a) (_,b) -> compare a b) binds)
  in

  (**** Check global variables ****)

  check_binds "global" globals;

  (**** Check functions ****)

  (* Collect function declarations for built-in functions: no bodies *)
  let built_in_decls =
    let add_bind map (ftyp, name, ty) = StringMap.add name {
```

```

    typ = ftyp;
    fname = name;
    formals = [(ty, "x")];
    locals = []; body = [] } map
in List.fold_left add_bind StringMap.empty [ (Void, "printi", Int);
      (Void, "prints", String);
      (Void, "printd", Double);
      (Void, "printb", Bool);
      (Int, "numPt", String);
      (Int, "totTime", File);
      (Double, "totDist", File);
      (Double, "avgSpeed", File);
      (File, "parseGPX", String);
      (Void, "printFirst", File);
      (Void, "stat", File);
      (Int, "intFloor", Double);
      (Double, "doubleFloor", Double);
      (Double, "squareRoot", Double);
      (Double, "getLat", Trackpoint);
      (Double, "getLon", Trackpoint);
      (Int, "getTime", Trackpoint);
      ]

in

(* Add function name to symbol table *)
let add_func map fd =
  let built_in_err = "function " ^ fd.fname ^ " may not be defined"
  and dup_err = "duplicate function " ^ fd.fname
  and make_err er = raise (Failure 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

```

```

(*let _ = find_func "main" in (* Ensure "main" is defined *)*)

let check_function func =
  (* Make sure no formals or locals are void or duplicates *)
  check_binds "formal" func.formals;
  check_binds "local" func.locals;

  (* Raise an exception if the given rvalue type cannot be assigned to
  the given lvalue type *)
  let check_assign lvaluet rvaluet err =
    if lvaluet = rvaluet then lvaluet else raise
      (Failure ("Assign error in semant " ^ err))
  in

  (* Build local symbol table of variables for this function *)
  let symbols = List.fold_left (fun m (ty, name) -> StringMap.add name ty m)
    StringMap.empty (globals @ func.formals @ func.locals)
  in

  (* Return a variable from our local symbol table *)
  let type_of_identifier s =
    try StringMap.find s symbols
    with Not_found -> raise (Failure ("undeclared identifier " ^ s))
  in

  (* Return a semantically-checked expression, i.e., with a type *)
  let rec expr = function
    Literal l -> (Int, SLiteral l)
  | DLit l -> (Double, SDLit l)
  | SLit l -> (String, SSLit l)
  | BLit l -> (Bool, SBLit l)
  | Noexpr -> (Void, SNoexpr)
  | Id s -> (type_of_identifier s, SId s)
  | TrackpointLit(e1, e2, e3) -> (* check lat, long, time are correct types *)
    let (t1, _) = expr e1
      and (t2, _) = expr e2
      and (t3, _) = expr e3 in

    if t1 = Double && t2 = Double && t3 = Int then
      (Trackpoint, STrackpointLit (expr e1, expr e2, expr e3))
    else raise (Failure ("expected numeric trackpoint args"))
  | TrackpointAccess(_,a) ->
    (match a with

```

```

    | "lat" | "lon" | "time" -> (Double, SDLit a)
    | _ -> raise(Failure ("error semant access type"))
  )
| Assign(var, e) as ex ->
  let lt = type_of_identifier var
  and (rt, e') = expr e in
  let err = "illegal assignment " ^ string_of_typ lt ^ " = " ^
    string_of_typ rt ^ " in " ^ string_of_expr ex
  in (check_assign lt rt err, SAssign(var, (rt, e')))
| Unop(op, e) as ex ->
  let (t, e') = expr e in
  let ty = match op with
    Neg when t = Int || t = Double -> t
  | Not when t = Bool -> Bool
  | _ -> raise (Failure ("illegal unary operator " ^
    string_of_uop op ^ string_of_typ t ^
    " in " ^ string_of_expr ex))
  in (ty, SUnop(op, (t, e')))
| Binop(e1, op, e2) as e ->
  let (t1, e1') = expr e1
  and (t2, e2') = expr e2 in
  (* All binary operators require operands of the same type *)
  let same = t1 = t2 in
  (* Determine expression type based on operator and operand types *)
  let ty = match op with
    Add | Sub | Mult | Div when same && t1 = Int -> Int
  | Add | Sub | Mult | Div when same && t1 = Double -> Double
  | Equal | Neq when same -> Bool
  | Less | Leq | Greater | Geq
    when same && (t1 = Int || t1 = Double) -> Bool
  | And | Or when same && t1 = Bool -> Bool
  | _ -> raise (
    Failure ("illegal binary operator " ^
      string_of_typ t1 ^ " " ^ string_of_op op ^ " " ^
      string_of_typ t2 ^ " in " ^ string_of_expr e))
  in (ty, SBinop((t1, e1'), op, (t2, e2')))
| Call(fname, args) as call ->
  let fd = find_func fname in
  let param_length = List.length fd.formals in
  if List.length args != param_length then
    raise (Failure ("expecting " ^ string_of_int param_length ^
      " arguments in " ^ string_of_expr call))
  else let check_call (ft, _) e =
    let (et, e') = expr e in
    let err = "illegal argument found " ^ string_of_typ et ^
      " expected " ^ string_of_typ ft ^ " in " ^ string_of_expr e

```

```

        in (check_assign ft et err, e')
    in
    let args' = List.map2 check_call fd.formals args
    in (fd.typ, SCall(fname, args'))
| ArrayLiteral(l) ->
    if List.length l > 0 then
        let typ = expr(List.nth l 0) in
        let (arraytype, _) = typ in
        let check_array e =
            let (et, e') = expr e in (et, e')
            in let l' = List.map check_array l in
            let types e =
                let (a, _) = expr e in
                if a <> arraytype then
                    raise(Failure("array of different types" ^
                        string_of_typ a ^ " is not " ^ string_of_typ arraytype))
                in let _ = List.map types l in
                (ArrayType(arraytype), SArrayLiteral(l'))
        else raise (Failure ("Must have at least one element in list upon initialization"))
| ArrayAccess (e1, e2) ->
    let (array, e1') = expr e1
    and (index, e2') = expr e2 in
    if index <> Int then raise(Failure("index must be integer"))
    else let check_arr_t array =
        match array with
        | ArrayType(t) -> t
        | _ -> raise(Failure("can only access arrays"))
    in (check_arr_t array, SArrayAccess((array, e1'), (index, e2')))
| ArrayAppend (e1, e2) ->
    let (array, e1') = expr e1
    and (typ, element) = expr e2 in
    let check_arr_t array = match array with
        | ArrayType(t) -> t
        | _ -> raise(Failure("can only append to arrays"))
    (*check same type here*)
    in (check_arr_t array, SArrayAppend((array, e1'), (typ, element)))
(*| TrackArrayLiteral(l) ->
    if List.length l > 0 then
        let siz = List.length l in
        let typ = expr(List.nth l 0) in
        let (arraytype, _) = typ in
        let check_array e =
            let (et, e') = expr e in (et, e')
            in let l' = List.map check_array l in
            let types e =

```

```

        let (a, _) = expr e in
        if a <> arraytype then
            raise(Failure("array of different types"))
        in let _ = List.map types l in
        (TrackArrayType(arraytype, List.length l'), STrackArrayLiteral(l'))

    else raise (Failure ("empty TrackArray initialization not supported"))
| TrackArrayAccess (e1, e2) as e ->
    let (tatype, e1') = expr e1 in
    let (index, e2') = expr e2 in
    if index <> Int then raise(Failure("index must be integer"))
    else let check_arr_t tatype =
        match tatype with
        | TrackArrayType(t, _) -> t
        | _ -> raise(Failure("can only access TrackArray")) in
    let get_size tatype =
        match tatype with
        | TrackArrayType(_, siz) -> siz
        | _ -> raise(Failure("no size in TrackArray type")) in

        (TrackArrayType(check_arr_t tatype, get_size tatype),
         STrackArrayAccess((tatype, e1'), (index, e2')))
| TrackArrayInit(i, s) ->
    let expr_id = expr i in
    let (typ, _) = expr_id in
    let check_type = match typ with
        | TrackArrayType(t, _) -> t
        | _ -> raise(Failure("can only initialize a TrackArrayType with new")) in
    let match_type = match check_type with
        | Trackpoint -> Trackpoint
        | _ -> raise(Failure("expected Trackpoint, got" ^ string_of_typ check_type)) in
    (TrackArrayType(match_type, s), STrackArrayInit(expr_id, (Int, SLiteral s))) *)
in

let check_bool_expr e =
    let (t', e') = expr e
    and err = "expected Boolean expression in " ^ string_of_expr e
    in if t' != Bool then raise (Failure err) else (t', e')
in

(* Return a semantically-checked statement i.e. containing sexprs *)
let rec check_stmt = function
    Expr e -> SExpr (expr e)
  | If(p, b1, b2) -> SIf(check_bool_expr p, check_stmt b1, check_stmt b2)
  | For(e1, e2, e3, st) ->

```

```

      SFor(expr e1, check_bool_expr e2, expr e3, check_stmt st)
| While(p,s) -> SWhile(check_bool_expr p, check_stmt s)
| Return e -> let (t, e') = expr e in
  if t = func.typ then SReturn (t, e')
  else raise (
    Failure ("return gives " ^ string_of_ttyp t ^ " expected " ^
      string_of_ttyp func.typ ^ " in " ^ string_of_expr e))

  (* A block is correct if each statement is correct and nothing
     follows any Return statement. Nested blocks are flattened. *)
| Block sl ->
  let rec check_stmt_list = function
    [Return _ as s] -> [check_stmt s]
  | Return _ :: _ -> raise (Failure "nothing may follow a return")
  | Block sl :: ss -> check_stmt_list (sl @ ss) (* Flatten blocks *)
  | s :: ss -> check_stmt s :: check_stmt_list ss
  | [] -> []
  in SBlock(check_stmt_list sl)

in (* body of check_function *)
{ styp = func.typ;
  sfname = func.fname;
  sformals = func.formals;
  slocals = func.locals;
  sbody = match check_stmt (Block func.body) with
    SBlock(sl) -> sl
  | _ -> raise (Failure ("internal error: block didn't become a block?"))
}

in (globals, List.map check_function functions)

```

10.8 codegen.ml

Authors: Ashley, Elisa, Katherine

```

(*
  Code generation for Gwiz:
  translate takes a semantically checked AST and produces LLVM IR
  Based on MicroC
  File: codegen.ml
  Authors: Ashley Kim, Elisa Luo, Katherine Duff
  PLT Spring 2021

```

LLVM tutorial: Make sure to read the OCaml version of the tutorial

<http://llvm.org/docs/tutorial/index.html>

Detailed documentation on the OCaml LLVM library:

<http://llvm.moe/>
<http://llvm.moe/ocaml/>

*)

```
module L = Lllvm
module A = Ast
open Sast

module StringMap = Map.Make(String)

(* translate : Sast.program -> Lllvm.module *)
let translate (globals, functions) =
  let context = L.global_context () in

  (* Create the LLVM compilation module into which
     we will generate code *)
  let the_module = L.create_module context "gWiz" in

  (* Get types from the context *)
  let i32_t = L.i32_type context
  and i8_t = L.i8_type context
  and double_t = L.double_type context
  and string_t = L.pointer_type (L.i8_type context)
  and i1_t = L.i1_type context
  and void_t = L.void_type context in
  (*Pointers for the array*)
  let i32_ptr_t = L.pointer_type i32_t
  and i64_t = L.i64_type context in
  let node_t = let node_t = L.named_struct_type context "Node" in
    L.struct_set_body node_t [| i64_t ; L.pointer_type node_t |] false;
    node_t in
  let node_pt = L.pointer_type node_t in
  let arr_t = let arr_t = L.named_struct_type context "Array" in
    L.struct_set_body arr_t [| node_pt ; i32_t |] false;
    arr_t in
  let arr_pt = L.pointer_type arr_t
  in

  let pointer_t = L.pointer_type in
  let trackpoint_t = L.named_struct_type context "trackpoint_t" in
```



```

L.struct_set_body trackpoint_t [| double_t; double_t; i32_t|] false;
let trk_pointer_t = pointer_t trackpoint_t in

(* Return the LLVM type for a gWiz type *)
let ltype_of_typ = function
  A.Int    -> i32_t
  | A.Double -> double_t
  | A.Void   -> void_t
  | A.String -> string_t
  | A.Bool   -> i1_t
  | A.Trackpoint -> trk_pointer_t
  | A.ArrayType(_) -> arr_pt
  | A.File    -> i32_ptr_t
in

      (*Create a mpa of global variables after creating each *)
      let global_vars: L.llvalue StringMap.t =
let global_var m (t,n) =
  let init = match t with
    (*add other types to pattern match here*)
    A.Double -> L.const_float (ltype_of_typ t) 0.0
    | A.Trackpoint -> L.const_pointer_null (ltype_of_typ t)
    | _ -> L.const_int (ltype_of_typ t) 0
  in StringMap.add n (L.define_global n init the_module) m in
List.fold_left global_var StringMap.empty globals in

(* Declare built in print functions *)
let printf_t : L.lltype =
  L.var_arg_function_type i32_t [| L.pointer_type i8_t |] in
let printf_func : L.llvalue =
  L.declare_function "printf" printf_t the_module in

(* Declare built in array functions *)
let initlist_t : L.lltype =
  L.var_arg_function_type arr_pt [| i64_t |] in
let initlist_func : L.llvalue =
  L.declare_function "initList" initlist_t the_module in

let addend_t : L.lltype =
  L.var_arg_function_type node_pt [| arr_pt; i64_t |] in
let addend_func : L.llvalue =
  L.declare_function "addEnd" addend_t the_module in

let getelemindex_t : L.lltype =
  L.var_arg_function_type i64_t [| arr_pt; i32_t |] in
let getelemindex_func : L.llvalue =

```

```

L.declare_function "getElemIndex" getelemindex_t the_module in

(*Declare GPX Parsing functions *)
let numpt_t: L.lltype =
  L.function_type i32_t [| string_t |] in
let numpt_func: L.llvalue =
  L.declare_function "numPt" numpt_t the_module in

let parseGPX_t: L.lltype =
  L.function_type i32_ptr_t [| string_t |] in
let parseGPX_func: L.llvalue =
  L.declare_function "parseGPX" parseGPX_t the_module in

let printFirst_t: L.lltype =
  L.function_type i32_t [| i32_ptr_t |] in
let printFirst_func: L.llvalue =
  L.declare_function "printFirst" printFirst_t the_module in

(*declare built in gpx functions*)
let totalTime_t: L.lltype =
  L.function_type i32_t [| i32_ptr_t |] in
let totalTime_func: L.llvalue =
  L.declare_function "totTime" totalTime_t the_module in

let totDist_t: L.lltype =
  L.function_type double_t [| i32_ptr_t |] in
let totDist_func: L.llvalue =
  L.declare_function "totDist" totDist_t the_module in

let avgSpeed_t: L.lltype =
  L.function_type double_t [| i32_ptr_t |] in
let avgSpeed_func: L.llvalue =
  L.declare_function "avgSpeed" avgSpeed_t the_module in

let stat_t: L.lltype =
  L.function_type i32_t [| i32_ptr_t |] in
let stat_func: L.llvalue =
  L.declare_function "stat" stat_t the_module in

(*declare built in math functions*)
let intFloor_t: L.lltype =
  L.function_type i32_t [| double_t |] in
let intFloor_func: L.llvalue =
  L.declare_function "intFloor" intFloor_t the_module in

```

```

let squareRoot_t: L.lltype =
  L.function_type double_t [| double_t |] in
let squareRoot_func: L.llvalue =
  L.declare_function "squareRoot" squareRoot_t the_module in

let doubleFloor_t: L.lltype =
  L.function_type double_t [| double_t |] in
let doubleFloor_func: L.llvalue =
  L.declare_function "doubleFloor" doubleFloor_t the_module in

(* Linked getter functions for Trackpoint *)
let getLat_t : L.lltype =
  L.function_type double_t [|trk_pointer_t|] in
let getLat_func : L.llvalue =
  L.declare_function "getLat" getLat_t the_module in

let getLon_t : L.lltype =
  L.function_type double_t [|trk_pointer_t|] in
let getLon_func : L.llvalue =
  L.declare_function "getLon" getLon_t the_module in

let getTime_t : L.lltype =
  L.function_type i32_t [|trk_pointer_t|] in
let getTime_func : L.llvalue =
  L.declare_function "getTime" getTime_t the_module in

(* Ensures int helper *)
let ensureInt c =
  if L.type_of c = double_t then (L.const_fptosi c i32_t) else c in

(* Ensures double helper *)
let ensureDouble c =
  if L.type_of c = double_t then c else (L.const_sitofp c double_t) in

(* Define each function (arguments and return type) so we can
   call it even before we've created its body *)
let function_decls : (L.llvalue * sfunc_decl) StringMap.t =
  let function_decl m fdecl =
    let name = fdecl.sfname
    and formal_types =
      Array.of_list (List.map (fun (t,_) -> ltype_of_typ t) fdecl.sformals)
    in let ftype = L.function_type (ltype_of_typ fdecl.styp) formal_types in
    StringMap.add name (L.define_function name ftype the_module, fdecl) m in
  List.fold_left function_decl StringMap.empty functions in

```

```

(* Fill in the body of the given function *)
let build_function_body fdecl =
  let (the_function, _) = StringMap.find fdecl.sfname function_decls in
  let builder = L.builder_at_end context (L.entry_block the_function) in

  let int_format_str = L.build_global_stringptr "%d\n" "fmt" builder
  and string_format_str = L.build_global_stringptr "%s\n" "fmt" builder
  and double_format_str = L.build_global_stringptr "%g\n" "fmt" builder
  and bool_format_str = L.build_global_stringptr "%d\n" "fmt" builder
  in

  (*Construct the function's "locals": formal arguments and
  locally declared variables. Allocate each on the stack,
  initialize their value, if appropriate, and remember
  their values in the "locals" map *)
  let local_vars =
    let add_formal m (t, n) p =
      let local = L.build_alloca (ltype_of_typ t) n builder in
      ignore (L.build_store p local builder);
      StringMap.add n local m
    and add_local m (t, n) =
      let local_var = L.build_alloca (ltype_of_typ t) n builder
      in StringMap.add n local_var m in

    let formals = List.fold_left2 add_formal StringMap.empty
      fdecl.sformals (Array.to_list (L.params the_function)) in
    List.fold_left add_local formals fdecl.slocals in

  (*Return the value for a variable or formal argument.
  Check local names first, then global names*)
  let lookup n = try StringMap.find n local_vars
    with Not_found -> StringMap.find n global_vars in

  (* Construct code for an expression; return its value *)
  let rec expr builder ((t, e) : sexpr) = match e with
    | SLiteral i -> L.const_int i32_t i
    | SDLit l -> L.const_float_of_string double_t l
    | SBLit b -> L.const_int i1_t (if b then 1 else 0)
    | SSLit s -> L.build_global_stringptr s "str" builder
    | SNoexpr -> L.const_int i32_t 0
    | SId s -> L.build_load (lookup s) s builder
    | STrackpointLit (e1, e2, e3) ->
      let tmp = L.build_alloca trackpoint_t "trackpoint_tmp" builder in
      let tptr = L.build_alloca (L.pointer_type trackpoint_t) "trackpoint_pointer"
      builder in

```

```

let e1' = ensureDouble (expr builder e1)
and e2' = ensureDouble (expr builder e2)
and e3' = ensureInt (expr builder e3) in

let lat_tmp = L.build_struct_gep ttmp 0 "lat" builder in
  ignore (L.build_store e1' lat_tmp builder);
let lon_tmp = L.build_struct_gep ttmp 1 "lon" builder in
  ignore (L.build_store e2' lon_tmp builder);
let time_tmp = L.build_struct_gep ttmp 2 "time" builder in
  ignore (L.build_store e3' time_tmp builder);
  ignore (L.build_store ttmp tptr builder);
let trkld = L.build_load tptr "" builder in trkld
| STrackpointAccess (e1, s2) ->
  let trackpoint = expr builder e1 in
  (match s2 with
  | "lat" -> L.build_call getLat_func [|trackpoint|] "getLat" builder
  | "lon" -> L.build_call getLon_func [|trackpoint|] "getLon" builder
  | "time" -> L.build_call getTime_func [|trackpoint|] "getTime" builder
  | _ -> raise (Failure ("Attribute does not exist")))
  )
| SArrayLiteral l -> build_arr builder t l
| SArrayAccess (array, index) ->
  let (t, _) = array in
  let check_arr_type = match t with
    A.ArrayType t -> t
    | _ -> raise(Failure("Can only access an array")) in
  let var = L.build_call getelemindex_func [| expr builder array;
    expr builder index|] "getElement" builder in
  (match check_arr_type with
    A.Int -> L.build_trunc var i32_t "arrayAcc" builder
    | A.Trackpoint -> L.build_inttoptr var trk_pointer_t "arrayAcc" builder
    | A.File -> L.build_inttoptr var i32_ptr_t "arrayAcc" builder
    | _ -> raise(Failure("Accessing illegal array type")))
  )
| SArrayAppend (array, element) ->
  let(t1, _) = array in
  let check_arr_type =
    match t1 with
    A.ArrayType t -> t
    | _ -> raise(Failure("can only append to an array type")) in
  let (elem_typ, _) = element in
  if check_arr_type <> elem_typ then raise(Failure("element type must be
  the same as array type"))
  else
  let built_expr = expr builder element in
  let t = L.type_of built_expr in

```

```

    let data = match L.classify_type t with
      L.TypeKind.Pointer -> L.build_pointercast built_expr i64_t "" builder
    | L.TypeKind.Integer -> L.build_zext built_expr i64_t "" builder
    | _ -> raise (Failure("cannot convert array type "))
  in L.build_call addend_func [| expr builder array; data |] "addEnd" builder

(*| STrackArrayLiteral(l) ->
  (*initialize array*)
  let list_size = List.length l in
  let ll_listsize = L.const_int i32_t list_size in
  let list_hd = (List.nth l 0) in
  let (ast_typ, other) = list_hd in
  let match_typ = match ast_typ with
    A.Trackpoint -> A.Trackpoint
  | _ -> raise(Failure("Track list must be Trackpoint")) in
  let init_array = L.build_array_malloc (trk_pointer_t) ll_listsize ""
    builder in

  (* add the list elements *)
  let elements = List.map (fun e -> expr builder e) l in
  ignore(List.fold_left (fun i elem ->
    let idx = L.const_int i32_t i in
    let elem_ptr = L.build_gep init_array [| idx |] "" builder in
    let a = L.build_store elem elem_ptr builder in
    i+1) 0 elements); init_array

| STrackArrayAccess (array, index) ->
  let arr_var = expr builder array in
  let idx = expr builder index in

  let ptr = L.build_gep arr_var [| idx |] "" builder in
  let cast_pt = L.build_ptrtoint ptr i64_t "" builder in
  let tpt = L.build_pointercast cast_pt trk_pointer_t "" builder in

  L.build_load tpt "" builder

| STrackArrayInit (tarrtype, siz) ->
  let len = expr builder siz in
  let init_array = L.build_array_malloc (trk_pointer_t) len "" builder in
  (* let init_array = L.build_array_alloc (trackpoint_t) len "" builder in *)
  init_array *)

| SAssign (s, e) -> let e' = expr builder e in
  ignore(L.build_store e' (lookup s) builder); e'
| SBinop ((A.Double,_) as e1, op, e2) ->

```

```

let e1' = expr builder e1
and e2' = expr builder e2 in
(match op with
  A.Add      -> L.build_fadd
| A.Sub      -> L.build_fsub
| A.Mult     -> L.build_fmuls
| A.Div      -> L.build_fdiv
| A.Equal    -> L.build_fcml L.Fcml.Oeq
| A.Neq     -> L.build_fcml L.Fcml.One
| A.Less    -> L.build_fcml L.Fcml.Olt
| A.Leq     -> L.build_fcml L.Fcml.Ole
| A.Greater -> L.build_fcml L.Fcml.Ogt
| A.Geq     -> L.build_fcml L.Fcml.Oge
| A.And | A.Or ->
    raise (Failure "internal error:
semant should have rejected and/or on double")
) e1' e2' "tmp" builder
| SBinop (e1, op, e2) ->
let e1' = expr builder e1
and e2' = expr builder e2 in
(match op with
  A.Add      -> L.build_add
| A.Sub      -> L.build_sub
| A.Mult     -> L.build_muls
| A.Div      -> L.build_sdiv
| A.And      -> L.build_and
| A.Or       -> L.build_or
| A.Equal    -> L.build_icml L.Icml.Eq
| A.Neq     -> L.build_icml L.Icml.Ne
| A.Less    -> L.build_icml L.Icml.Slt
| A.Leq     -> L.build_icml L.Icml.Sle
| A.Greater -> L.build_icml L.Icml.Sgt
| A.Geq     -> L.build_icml L.Icml.Sge
) e1' e2' "tmp" builder
| SUnop(op, ((t, _) as e)) ->
let e' = expr builder e in
  (match op with
    A.Neg when t = A.Double -> L.build_fneg
  | A.Neg                    -> L.build_neg
  | A.Not                    -> L.build_not) e' "tmp" builder
| SCall ("printf", [e]) ->
  L.build_call printf_func [| int_format_str ; (expr builder e) |]
  "printf" builder
| SCall ("prints", [e]) ->
  L.build_call printf_func [| string_format_str ; (expr builder e) |]
  "printf" builder

```

```

| SCall ("printf", [e]) ->
    L.build_call printf_func [| double_format_str ; (expr builder e) |]
      "printf" builder
| SCall ("printfb", [e]) ->
    L.build_call printf_func [| bool_format_str ; (expr builder e) |]
      "printf" builder
| SCall ("numPt", [e]) ->
    L.build_call numpt_func [| (expr builder e) |] "numPt" builder
| SCall ("printFirst", [e]) ->
    L.build_call printFirst_func [| (expr builder e) |] "printFirst" builder
| SCall ("stat", [e]) ->
    L.build_call stat_func [| (expr builder e) |] "stat" builder
| SCall ("totTime", [e]) ->
    L.build_call totTime_func [| (expr builder e) |] "totTime" builder
| SCall ("totDist", [e]) ->
    L.build_call totDist_func [| (expr builder e) |] "totDist" builder
| SCall ("avgSpeed", [e]) ->
    L.build_call avgSpeed_func [| (expr builder e) |] "avgSpeed" builder
| SCall ("intFloor", [e]) ->
    L.build_call intFloor_func [| (expr builder e) |] "intFloor" builder
| SCall ("squareRoot", [e]) ->
    L.build_call squareRoot_func [| (expr builder e) |] "squareRoot" builder
| SCall ("doubleFloor", [e]) ->
    L.build_call doubleFloor_func [| (expr builder e) |] "doubleFloor" builder
| SCall ("parseGPX", [e]) ->
    L.build_call parseGPX_func [| (expr builder e) |] "parseGPX" builder
| SCall("getLat", [e]) ->
    L.build_call getLat_func [| (expr builder e) |] "getLat" builder
| SCall("getLon", [e]) ->
    L.build_call getLon_func [| (expr builder e) |] "getLon" builder
| SCall("getTime", [e]) ->
    L.build_call getTime_func [| (expr builder e) |] "getTime" builder
| SCall (f, args) ->
    let (fdef, fdecl) = StringMap.find f function_decls in
    let llargs = List.rev (List.map (expr builder) (List.rev args)) in
    let result = (match fdecl.styp with
        A.Void -> ""
        | _ -> f ^ "_result") in
    L.build_call fdef (Array.of_list llargs) result builder

```

(Functions to initialize and build Array Literal*)*

```

and element_type = function
  A.ArrayType t -> t
  | _ -> raise (Failure("need array type"))

```



```

and element_size t =
  let typ_size = element_type t in
  L.size_of(ltype_of_typ typ_size)

and build_arr builder t arr =
  let type_size = element_size t in
  let list = L.build_call initlist_func [| type_size |] "initList" builder in

  let add_end e =
    let built_expr = expr builder e in
    let t = L.type_of built_expr in
    let data = match L.classify_type t with
      L.TypeKind.Pointer -> L.build_pointercast built_expr i64_t "" builder
    | L.TypeKind.Integer -> L.build_zext built_expr i64_t "" builder
    | _ -> raise (Failure("cannot convert array type"))
    in
    ignore(L.build_call addend_func [| list; data |] "addEnd" builder)
  in List.iter add_end arr; list
in

(* LLVM insists each basic block end with exactly one "terminator"
   instruction that transfers control. This function runs "instr builder"
   if the current block does not already have a terminator. Used,
   e.g., to handle the "fall off the end of the function" case. *)
let add_terminal builder instr =
  match L.block_terminator (L.insertion_block builder) with
  Some _ -> ()
  | None -> ignore (instr builder) in

(* Build the code for the given statement; return the builder for
   the statement's successor (i.e., the next instruction will be built
   after the one generated by this call) *)

let rec stmt builder = function
  SBlock sl -> List.fold_left stmt builder sl
  | SExpr e -> ignore(expr builder e); builder
  | SReturn e -> ignore(match fdecl.styp with
    (* Special "return nothing" instr *)
    A.Void -> L.build_ret_void builder
    (* Build return statement *)
    | _ -> L.build_ret (expr builder e) builder );
    builder
  | SIf (predicate, then_stmt, else_stmt) ->
    let bool_val = expr builder predicate in
    (* Add merge block to the list of blocks for the function *)

```

```

        let merge_bb = L.append_block context "merge" the_function in
    (* Partial function to create branch with merge block *)
    let build_br_merge = L.build_br merge_bb in

    (* "then" block *)
    let then_bb = L.append_block context "then" the_function in
    add_terminal (stmt (L.builder_at_end context then_bb) then_stmt)
    build_br_merge;

    (* "else block" *)
    let else_bb = L.append_block context "else" the_function in
    add_terminal (stmt (L.builder_at_end context else_bb) else_stmt)
    build_br_merge;

    ignore(L.build_cond_br bool_val then_bb else_bb builder);
    L.builder_at_end context merge_bb

| SWhile (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 (L.builder_at_end context body_bb) body)
        (L.build_br pred_bb);

    let pred_builder = L.builder_at_end context pred_bb in
    let bool_val = expr 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

(* Implement for loops as while loops *)
| SFor (e1, e2, e3, body) -> stmt builder
    ( SBlock [SExpr e1 ; SWhile (e2, SBlock [body ; SExpr e3]) ] )

in

(* Build the code for each statement in the function *)
let builder = stmt builder (SBlock fdecl.sbody) in

(* Add a return if the last block falls off the end *)
add_terminal builder (match fdecl.styp with
    A.Void -> L.build_ret_void
    | A.Double -> L.build_ret(L.const_float double_t 0.0)
    | A.Int -> L.build_ret (L.const_int i32_t 0)

```

```

        | t -> L.build_ret (L.const_int (ltype_of_typ t) 0))
in

List.iter build_function_body functions;
the_module

```

10.9 struct.c

Author: Ashley

```

1 #include <stdio.h>
2 #include "struct.h"
3
4 double getLat(Trackpoint *t){
5     struct Trackpoint *track = (struct Trackpoint *) t;
6     return t->lat;
7 }
8
9 double getLon(Trackpoint *t){
10    struct Trackpoint *track = (struct Trackpoint *) t;
11    return t->lon;
12 }
13
14 int getTime(Trackpoint *t){
15    struct Trackpoint *track = (struct Trackpoint *) t;
16    return t->time;
17 }
18
19 #ifdef BUILD_TEST
20 int main(){
21    printf("in c main func \n");
22    struct Trackpoint t = {3.5, 2.3, 4};
23
24    double d = getLat(&t);
25    printf("lat is %f \n", d);
26
27    return 0;
28 }
29 #endif

```

10.10 struct.h

Author: Ashley

```

1 #ifndef STRUCT_H
2 #define STRUCT_H
3
4 typedef struct Trackpoint {
5     double lat;

```

```

6     double lon;
7     int time;
8 } Trackpoint;
9
10
11 #endif

```

10.11 list.c

Author: Katherine

```

1  /* Create a simple linked list that can be initialized,
2  * accessed, and have elements appended to the end
3  *
4  */
5
6  #include <stdio.h>
7  #include <stdlib.h>
8
9
10 // Create a node
11 struct Node {
12     unsigned long data;
13     struct Node *next;
14 };
15
16 // Create the linked list, head is the first element
17 struct List {
18     struct Node *head;
19     size_t size_of_type;
20 };
21
22 // Initialize an empty list
23 struct List *initList(size_t size_of_type) {
24     struct List *list = malloc(sizeof(struct List));
25     if (list == NULL) {
26         perror("failed initialization");
27         exit(1);
28     }
29     list->head = NULL;
30     list->size_of_type = size_of_type;
31     return list;
32 }
33
34 // Access element at specified index, return the data, -1 if out of
35 // bounds
36 unsigned long getElemIndex(struct List *list, int idx) {
37     struct Node *node = list->head;
38     int curIdx = 0;

```

```

38     while(node) {
39         if (curIdx == idx) {
40             return node->data;
41         }
42         curIdx++;
43         node = node->next;
44     }
45     return -1;
46 }
47
48 // Append element data to the end of list
49 struct Node *addEnd(struct List *list, unsigned long data) {
50
51     // empty list
52     if (list->head == NULL) {
53         struct Node *h = malloc(sizeof(struct Node));
54         h->data = data;
55         list->head = h;
56         return h;
57     } else {
58         // find the end of the list
59         struct Node *node = list->head;
60         while (node->next) {
61             node = node->next;
62         }
63
64         struct Node *toAdd = malloc(sizeof(struct Node));
65         if (toAdd == NULL) {
66             perror("create node failed");
67             exit(1);
68         }
69         node->next = toAdd;
70         toAdd->data = data;
71         return toAdd;
72     }
73 }
74
75
76
77
78 #ifdef BUILD_TEST
79 int main(){
80
81 }
82 #endif

```

10.12 parse_gpx.c

Authors: Elisa, Rebecca

```

1 #include <stdio.h>
2 #include <string.h>
3 #include <stdlib.h>
4 #include <libxml/xmlmemory.h>
5 #include <libxml/parser.h>
6 #include <libxml/xpath.h>
7 #include <libxml/xpathInternals.h>
8
9 // #define __USE_XOPEN // For strttime
10 #include <time.h>
11
12 #include <math.h>
13 #define pi 3.14159265358979323846
14
15
16 /*
17 Description for a track point
18 */
19
20 struct track_point {
21     double lat, lon;
22     int time;
23 };
24
25 double deg2rad(double deg) {
26     return (deg * pi / 180);
27 }
28
29 double rad2deg(double rad) {
30     return (rad * 180 / pi);
31 }
32
33 /*
34 credit to: https://www.geodatasource.com/developers/c
35 for the distance function
36 */
37 double distance(double lat1, double lon1, double lat2, double lon2, char
    unit) {
38     double theta, dist;
39     if ((lat1 == lat2) && (lon1 == lon2)) {
40         return 0;
41     }
42     else {
43         theta = lon1 - lon2;
44         dist = sin(deg2rad(lat1)) * sin(deg2rad(lat2)) + cos(deg2rad(lat1))
            * cos(deg2rad(lat2)) * cos(deg2rad(theta));
45         dist = acos(dist);
46         dist = rad2deg(dist);
47         dist = dist * 60 * 1.1515;
48         switch(unit) {

```

```

49     case 'M':
50         break;
51     case 'K':
52         dist = dist * 1.609344;
53         break;
54     case 'N':
55         dist = dist * 0.8684;
56         break;
57     }
58     return (dist);
59 }
60 }
61
62 int intFloor(double d){
63     return (int)d;
64 }
65
66 double squareRoot(double d){
67     return sqrt(d);
68 }
69
70 double doubleFloor(double d){
71     return floor(d);
72 }
73
74 // https://stackoverflow.com/questions/26376396/
75 // library-function-to-extract-substring-by-position-index
76 char * substring(char *c, int startint, int endint){
77     char *start = &c[startint];
78     char *end = &c[endint];
79     // Note the + 1 here, to have a null terminated substring
80     char *substr = (char *)calloc(1, end - start + 1);
81     memcpy(substr, start, end - start);
82     return substr;
83 }
84
85 // https://www.epochconverter.com/programming/c
86 int convert_time(char *timestr){
87     // printf("time in convert %s\n", timestr);
88     int year = strtol(substring(timestr, 0, 4), NULL, 0);
89     int month = strtol(substring(timestr, 5, 7), NULL, 0);
90     int mday = strtol(substring(timestr, 8, 10), NULL, 0);
91     int hour = strtol(substring(timestr, 11, 13), NULL, 0);
92     int min = strtol(substring(timestr, 14, 16), NULL, 0);
93     int sec = strtol(substring(timestr, 17, 19), NULL, 0);
94     struct tm t;
95     time_t t_of_day;
96
97     t.tm_year = year-1900; // Year - 1900
98     t.tm_mon = month;     // Month, where 0 = jan

```

```

99     t.tm_mday = mday;           // Day of the month
100    t.tm_hour = hour;
101    t.tm_min = min;
102    t.tm_sec = sec;
103    t.tm_isdst = -1;           // Is DST on? 1 = yes, 0 = no, -1 = unknown
104    t_of_day = mktime(&t);
105
106    // printf("seconds since the Epoch: %ld\n", (long) t_of_day);
107    return t_of_day;
108 }
109
110 xmlXPathObjectPtr getnodeset (xmlDocPtr doc, xmlChar *xpath,
111     xmlXPathContextPtr context){
112     xmlXPathObjectPtr result;
113
114     if (context == NULL) {
115         printf("Error in xmlXPathNewContext\n");
116         return NULL;
117     }
118     result = xmlXPathEvalExpression(xpath, context);
119     if (result == NULL) {
120         printf("Error in xmlXPathEvalExpression\n");
121         xmlXPathFreeObject(result);
122         return NULL;
123     }
124     if(xmlXPathNodeSetIsEmpty(result->nodesetval)){
125         xmlXPathFreeObject(result);
126         printf("No result\n");
127         return NULL;
128     }
129     return result;
130 }
131
132 void print_tps(struct track_point *tps, int tps_size) {
133     printf("(");
134     for (int j=0; j < tps_size; j++) {
135         printf("%f, %f, %d)", tps[j].lat, tps[j].lon, tps[j].time);
136     }
137     printf(")\n");
138 }
139
140 int sizeTp(struct track_point *tps) {
141     int s = sizeof(tps);
142     return s;
143 }
144
145 void printFirst(struct track_point *tps){
146     printf("%f, %f, %d \n", tps[0].lat, tps[0].lon, tps[0].time);
147 }
148
149 double totDist(struct track_point *tps) {

```



```

148     //returns total distance in km
149     int tps_size = tps[0].time; //retrieve size
150     double tot_dist = 0;
151     //printf("%ld\n", sizeof(tps));
152     for (int j=1; j < tps_size; j++) {
153         double tmp = distance(tps[j].lat, tps[j].lon, tps[j+1].lat,
154                               tps[j+1].lon, 'K');
155         tot_dist+=tmp;
156     }
157     return tot_dist;
158 }
159 int totTime(struct track_point *tps) {
160     //returns total time in mins
161     int tps_size = tps[0].time;
162     int tot_time = 0;
163     int startTime = tps[1].time;
164     int endTime = tps[tps_size].time;
165     tot_time = endTime-startTime;
166
167     return tot_time/60;
168 }
169 double avgSpeed(struct track_point *tps){
170     //returns avg. speed in km/h
171     double dist = totDist(tps);
172     double time = (double)totTime(tps); //in hours
173     time = time/60.0;
174     //printf("%f\n", time);
175     return dist/time;
176 }
177
178 void stat(struct track_point *tps){
179     printf("***** GPX FILE SUMMARY *****\n");
180     printf("Distance Covered: %f km\n", totDist(tps));
181     printf("Elapsed Time: %d min\n", totTime(tps));
182     printf("Average Speed: %f km/h\n", avgSpeed(tps));
183     printf("*****\n");
184 }
185
186 struct track_point * parseGPX(char *filepath) {
187     // Load file
188     xmlDocPtr doc = xmlParseFile(filepath);
189
190     if (doc == NULL) {
191         fprintf(stderr, "Document not parsed successfully.");
192     }
193     xmlNodePtr cur = xmlDocGetRootElement(doc);
194
195     if (cur == NULL) {
196         fprintf(stderr, "empty document");

```

```

197     xmlFreeDoc(doc);
198     return 0;
199 }
200
201 xmlChar *xpath = (xmlChar *) "//gpx:trk/gpx:trkseg/gpx:trkpt";
202 xmlChar *timexpath = (xmlChar*)
203     "//gpx:trk/gpx:trkseg/gpx:trkpt/gpx:time";
204 xmlXPathContextPtr context = xmlXPathNewContext(doc);
205 xmlXPathRegisterNs(context, (xmlChar*)"gpx",
206     (xmlChar*)"http://www.topografix.com/GPX/1/1");
207
208 xmlNodeSetPtr nodeset;
209 xmlNodeSetPtr timenodeset;
210 xmlXPathObjectPtr result;
211 xmlXPathObjectPtr timeset;
212 int i = 0;
213 //xmlChar *keyword;
214
215 struct track_point *tps;
216
217 result = getnodeset(doc, xpath, context);
218 timeset = getnodeset(doc, timexpath, context);
219 xmlXPathFreeContext(context);
220
221 if (result) {
222     nodeset = result->nodesetval;
223     int size = nodeset->nodeNr;
224
225     // printf("size %d", size);
226     tps = (struct track_point *)malloc(sizeof(struct
227         track_point)*(size)+2);
228
229     //stash away size
230     struct track_point s;
231     s.lon = s.lat = 0.0;
232     s.time = size;
233     tps[0] = s;
234
235     timenodeset = timeset->nodesetval;
236     for (i=0; i < nodeset->nodeNr; i++) {
237
238         xmlNodePtr ptNode = result->nodesetval->nodeTab[i];
239
240         struct track_point tp;
241
242         xmlChar *lat = xmlGetProp(ptNode, (xmlChar*)"lat");
243         if (lat != 0) {
244             //printf("lat %s", lat);
245             tp.lat = strtod((char *)lat, NULL);
246             //fix

```

```

244         //printf("tp.lat %f", tp.lat);
245         xmlFree(lat);
246     }
247     xmlChar *lon = xmlGetProp(ptNode, (xmlChar*)"lon");
248     if (lat != 0) {
249         //printf("lon %s ", lon);
250         tp.lon = strtoc((char *)lon, NULL);
251         xmlFree(lon);
252     }
253     xmlChar *time = xmlNodeListGetString(doc,
254         timenodeset->nodeTab[i]->xmlChildrenNode, 1);
255     if (time != 0) {
256         // printf("time %s ", time);
257         int time_store;
258         time_store = convert_time((char *) time);
259         //printf("converted time %d\n", time_store);
260         tp.time = time_store;
261         xmlFree(time);
262     }
263     tps[i+1] = tp;
264     //printf(" testing %f %f %d\n", tps[i].lat, tps[i].lon,
265         tps[i].time);
266 }
267 xmlXPathFreeObject(timeset);
268 xmlXPathFreeObject(result);
269 //printf("Printing array of track_points: \n");
270 //print_tps(tps, size);
271 xmlFreeDoc(doc);
272 xmlCleanupParser();
273 return tps;
274 }
275 else {
276     return NULL;
277 }
278 }
279 int numPt(char *filepath) {
280     xmlDocPtr doc = xmlParseFile(filepath);
281
282     if (doc == NULL) {
283         fprintf(stderr, "Document not parsed successfully.");
284     }
285     xmlNodePtr cur = xmlDocGetRootElement(doc);
286
287     if (cur == NULL) {
288         fprintf(stderr, "empty document");
289         xmlFreeDoc(doc);
290         return 0;
291     }

```

```

292
293 xmlChar *xpath = (xmlChar *) "//gpx:trk/gpx:trkseg/gpx:trkpt";
294 xmlXPathContextPtr context = xmlXPathNewContext(doc);
295 xmlXPathRegisterNs(context, (xmlChar*)"gpx",
    (xmlChar*)"http://www.topografix.com/GPX/1/1");
296
297 xmlNodeSetPtr nodeset;
298 xmlXPathObjectPtr result;
299
300 result = getnodeset(doc, xpath, context);
301 xmlXPathFreeContext(context);
302 int size = 0;
303
304 if (result) {
305     nodeset = result->nodesetval;
306     size = nodeset->nodeNr;
307 }
308 xmlXPathFreeObject(result);
309 xmlFreeDoc(doc);
310 xmlCleanupParser();
311 return size;
312 }
313
314
315
316 #ifdef BUILDTEST
317 int main(int argc, char *argv[]){
318     if (argc < 2) {
319         printf("Wrong number of arguments");
320     }
321     printf("Number of track points: %d\n", numPt(argv[1]));
322     struct track_point * tps = parseGPX(argv[1]);
323     double dist = totDist(tps);
324     printf("size: %d\n", tps[0].time);
325     printf("total distance: %f\n", dist);
326     int time = totTime(tps);
327     printf("total time: %d\n", time);
328     double speed = avgSpeed(tps);
329     printf("avg speed: %f\n", speed);
330
331
332     //free(tps);
333 }
334 #endif

```

10.13 Makefile

Authors: Ashley, Rebecca

```

1 # run "make test" to compile everything and run the tests after make
   gwiz.native
2 .PHONY : test
3 test : all testall.sh
4   ./testall.sh
5
6 # run "make all" to build the executable
7 .PHONY : all
8 all: gwiz.native struct.o list.o parse_gpx.o
9
10 gwiz.native:
11   opam config exec -- \
12   ocamlbuild -use-ocamlfind gwiz.native
13
14 # run "make clean" to remove all generated files
15 .PHONY : clean
16 clean :
17   ocamlbuild -clean
18   rm -rf testall.log ocamlllvm *.diff *.ll *.exe *.s
19   rm -rf *.o *.out
20
21 # make gpx parser binary file
22
23 parse_gpx.o: parse_gpx.c
24   cc `xml2-config --cflags` -g -Wall "-I/usr/local/opt/libxml2/include"
   -c -o parse_gpx.o parse_gpx.c
25
26 struct: struct.c
27   cc -o struct.o -DBUILD_TEST struct.c
28 list: list.c
29   cc -o list -DBUILD_TEST list.c
30
31 TESTS = \
32   arith1 arith2 arith3 arith4 array1 array2 \
33   array3 arrayaccess double double 1 double 2 \
34   func func2 func3 func4 func5 func6 gcd1 gcd2 \
35   global1 global2 global3 gpx gpx1 gpx2 gpx3 \
36   helloworld if1 if2 if3 if4 if5 if6 local1 local2 \
37   math neg ops1 ops2 string1 struct struct1 var 1 \
38   var2 while1 while2 \
39
40 FAILS = \
41   array1 array2 array3 array4 array 5 array 6 \
42   assign1 double1 expr file for1 for2 for3 \
43   func1 func2 func3 func4 global1 global2 \
44   if1 if2 print1 return struct1 struct2 var 1\
45   while1 \
46
47 TESTFILES = $(TESTS:%=test-%.gw) $(TESTS:%=test-%.out) \
48   $(FAILS:%=fail-%.gw) $(FAILS:%=fail-%.err)

```

```

49
50 #TARFILES = ast.ml sast.ml codegen.ml Makefile _tags gwiz.ml
    gwizparse.mly \
51 # README scanner.mll semant.ml testall.sh \
52 # Dockerfile \
53 # $(TESTFILES:%=tests/%)

```

10.14 testall.sh

Authors: Ashley, Rebecca

```

1 # Regression testing script for GWiz
2 # Taken largely from the MicroC script
3
4
5 # Path to the LLVM interpreter
6 LLI="lli"
7 #LLI="/usr/local/opt/llvm/bin/lli"
8
9 # Path to the LLVM compiler
10 LLC="/usr/bin/llc"
11
12 # Path to the C compiler
13 CC="/usr/bin/cc"
14
15 # Path to the gwiz compiler.
16 GWIZ="./gwiz.native"
17 #GWIZ="_build/gwiz.native"
18
19
20 # Set time limit for all operations
21 ulimit -t 30
22
23 globallog=testall.log
24 rm -f $globallog
25 error=0
26 globalerror=0
27
28 keep=0
29
30 Usage() {
31     echo "Usage: testall.sh [options] [.gw files]"
32     echo "-k Keep intermediate files"
33     echo "-h Print this help"
34     exit 1
35 }
36
37 SignalError() {
38     if [ $error -eq 0 ] ; then

```

```

39     echo "FAILED"
40     error=1
41     fi
42     echo " $1"
43 }
44
45 # Compare <outfile> <reffile> <difffile>
46 # Compares the outfile with reffile. Differences, if any, written to
    difffile
47 Compare() {
48     generatedfiles="$generatedfiles $3"
49     echo diff -b $1 $2 ">" $3 1>&2
50     diff -b "$1" "$2" > "$3" 2>&1 || {
51     SignalError "$1 differs"
52     echo "FAILED $1 differs from $2" 1>&2
53     }
54 }
55
56 # Run <args>
57 # Report the command, run it, and report any errors
58 Run() {
59     echo $* 1>&2
60     eval $* || {
61     SignalError "$1 failed on $*"
62     return 1
63     }
64 }
65
66 # RunFail <args>
67 # Report the command, run it, and expect an error
68 RunFail() {
69     echo $* 1>&2
70     eval $* && {
71     SignalError "failed: $* did not report an error"
72     return 1
73     }
74     return 0
75 }
76
77 Check() {
78     error=0
79     basename=`echo $1 | sed 's/.*\\\/\\\/
    s/.gw//'\`
80
81     reffile=`echo $1 | sed 's/.gw$//'\`
82     basedir="`echo $1 | sed 's/\/[^\\/]*$//'\`/."
83
84     echo -n "$basename..."
85
86     echo 1>&2
87     echo "##### Testing $basename" 1>&2

```

```

88
89 generatedfiles=""
90
91 generatedfiles="$generatedfiles ${basename}.ll ${basename}.s
    ${basename}.exe ${basename}.out" &&
92 Run "$GWIZ" "$1" ">" "${basename}.ll" &&
93 Run "$LLC" "-relocation-model=pic" "${basename}.ll" ">"
    "${basename}.s" &&
94 Run "$CC" "-o" "${basename}.exe" "${basename}.s" "struct.o" "list.o"
    "parse_gpx.o" "\-L/usr/local/opt/libxml2/lib\" "\-xml2-config
    --libs\" "-lm"&&
95 Run "./${basename}.exe" > "${basename}.out"
96 Compare ${basename}.out ${reffile}.out ${basename}.diff &&
97 #./${basename}.exe
98
99 # Report the status and clean up the generated files
100
101 if [ $error -eq 0 ] ; then
102 if [ $keep -eq 0 ] ; then
103     rm -f $generatedfiles
104 fi
105 echo "OK"
106 echo "##### SUCCESS" 1>&2
107 else
108 echo "##### FAILED" 1>&2
109 globalerror=$error
110 fi
111 }
112
113 CheckFail() {
114     error=0
115     basename=`echo $1 | sed 's/.*\\\/\//
116                 s/.gw//'\`
117     reffile=`echo $1 | sed 's/.gw$//'\`
118     basedir=`echo $1 | sed 's/\[^\/\]*$//'\`/."
119
120     echo -n "$basename..."
121
122     echo 1>&2
123     echo "##### Testing $basename" 1>&2
124
125     generatedfiles=""
126
127     generatedfiles="$generatedfiles ${basename}.err ${basename}.diff" &&
128     RunFail "$GWIZ" "<" $1 "2>" "${basename}.err" ">>" $globallog &&
129     Compare ${basename}.err ${reffile}.err ${basename}.diff
130
131     # Report the status and clean up the generated files
132
133     if [ $error -eq 0 ] ; then

```



```

134     if [ $keep -eq 0 ] ; then
135         rm -f $generatedfiles
136     fi
137     echo "OK"
138     echo "##### SUCCESS" 1>&2
139     else
140     echo "##### FAILED" 1>&2
141     globalerror=$error
142     fi
143 }
144
145 while getopts kdpsh c; do
146     case $c in
147     k) # Keep intermediate files
148         keep=1
149         ;;
150     h) # Help
151         Usage
152         ;;
153     esac
154 done
155
156 shift `expr $OPTIND - 1`
157
158 LLIFail() {
159     echo "Could not find the LLVM interpreter \"$LLI\"."
160     echo "Check your LLVM installation and/or modify the LLI variable in
161     testall.sh"
162     exit 1
163 }
164
165 which "$LLI" >> $globallog || LLIFail
166
167 if [ $# -ge 1 ]
168 then
169     files=$@
170 else
171     files="tests/test-*.gw tests/fail-*.gw"
172 fi
173
174 for file in $files
175 do
176     case $file in
177     *test-*)
178         Check $file 2>> $globallog
179         ;;
180     *fail-*)
181         CheckFail $file 2>> $globallog
182         ;;
183     *)

```

```

183     echo "unknown file type $file"
184     globalerror=1
185     ;;
186     esac
187 done
188
189 exit $globalerror

```

10.15 Test Suite

ADD TESTS. alphabetical order pls u can get the order from github or in ur file dir it's alphabetized. also reminder to update makefile w alphabetized tests

10.15.1 Fail Test Files

fail-array1.gw

```

1 int main() {
2     int[] a;
3
4     a = [];
5 }
6
7 Fatal error: exception Failure("Must have at least one element in list
    upon initialization")

```

fail-array2.gw

```

1 int main() {
2     double[] a;
3     double b;
4
5     a = [1.0, 2.0];
6     b = a-<0>;
7
8 }
9
10 Fatal error: exception Failure("cannot convert array type")

```

fail-array3.gw

```

1 int main() {
2     int a;
3
4     a +<1>;
5     return 0;
6 }
7
8 Fatal error: exception Failure("can only append to arrays")

```

fail-array4.gw

```
1 int main() {
2     int[] a;
3     Trackpoint b;
4
5     b = (1.0, 2.0, 3);
6
7     a +<b>;
8     return 0;
9 }
10
11 Fatal error: exception Failure("element type must be the same as array
    type")
```

fail-array5.gw

```
1 int main() {
2     double[] d;
3     double a;
4     d = [1.0];
5
6     a = d-<0>;
7
8
9     return 0;
10 }
11
12 Fatal error: exception Failure("cannot convert array type")
```

fail-array6.gw

```
1 int main() {
2     int a;
3     int b;
4
5     a = b-<0>;
6     return 0;
7 }
8
9 Fatal error: exception Failure("Assign error in semant illegal
    assignment int = bool in i = false")
```

fail-assign1.gw

```
1 int main()
2 {
3     int i;
4     bool b;
5
6     i=3;
```

```

7   b = true;
8
9   i=false;
10  return 0;
11 }
12
13 Fatal error: exception Failure("Assign error in semant int=bool")

fail-double1.gw

1  int main()
2  {
3    -3.5 && 2.5; /* Float with AND? */
4    return 0;
5  }
6
7  Fatal error: exception Failure("illegal binary operator double && double
   in -3.5 && 2.5")

fail-expr.gw

1  int a;
2  bool b;
3
4  void foo(int c, bool d)
5  {
6    int dd;
7    bool e;
8    a + c;
9    c - a;
10   a * 3;
11   c / 2;
12   d + a; /* Error: bool + int */
13 }
14
15 int main()
16 {
17   return 0;
18 }
19
20 Fatal error: exception Failure("illegal binary operator bool + int in d
   + a")

fail-file.gw

1  int main(){
2
3    int i;
4    file f;
5
6    i=4;

```

```

7     f=4; /* invalid type */
8
9     return 0;
10  }
11
12  Fatal error: exception Failure("Assign error in semant illegal
    assignment file = int in f = 4")

```

fail-for1.gw

```

1  int main()
2  {
3      int i;
4
5      for (i = 0 ; i < 10 ; i = i + 1) {
6          if (i == 3) return 42;
7      }
8
9      for (j = 0; i < 10 ; i = i + 1) {} /* j undefined */
10
11     return 0;
12 }
13
14 Fatal error: exception Failure("undeclared identifier j")

```

fail-for2.gw

```

1  int main()
2  {
3      int i;
4
5      for (i = 0; j < 10 ; i = i + 1) {} /* j undefined */
6
7      return 0;
8  }
9
10 Fatal error: exception Failure("undeclared identifier j")

```

fail-for3.gw

```

1  int main()
2  {
3      int i;
4
5      for (i = 0; i ; i = i + 1) {} /* i is an integer, not Boolean */
6
7      return 0;
8  }
9
10 Fatal error: exception Failure("expected Boolean expression in i")

```

fail-func1.gw

```
1 int foo() {}
2
3 int bar() {}
4
5 int baz() {}
6
7 void bar() {} /* Error: duplicate function bar */
8
9 int main()
10 {
11     return 0;
12 }
13
14 Fatal error: exception Failure("duplicate function bar")
```

fail-func2.gw

```
1 int foo(int a, bool b, int c) { }
2
3 void bar(int a, bool b, int a) {} /* Error: duplicate formal a in bar */
4
5 int main()
6 {
7     return 0;
8 }
9
10 Fatal error: exception Failure("duplicate formal a")
```

fail-func3.gw

```
1 int foo(int a, bool b, int c) { }
2
3 void bar(int a, void b, int c) {} /* Error: illegal void formal b */
4
5 int main()
6 {
7     return 0;
8 }
9
10 Fatal error: exception Failure("illegal void formal b")
```

fail-func4.gw

```
1 int foo() {}
2
3 void bar() {}
4
5 int printi() {} /* Should not be able to define print */
6
```

```

7 void baz() {}
8
9 int main()
10 {
11     return 0;
12 }
13
14 Fatal error: exception Failure("function printi may not be defined")

fail-global1.gw

1 int c;
2 bool b;
3 void a; /* global variables should not be void */
4
5
6 int main()
7 {
8     return 0;
9 }
10
11 Fatal error: exception Failure("illegal void global a")

fail-global2.gw

1 int b;
2 bool c;
3 int a;
4 int b; /* Duplicate global variable */
5
6 int main()
7 {
8     return 0;
9 }
10
11 Fatal error: exception Failure("duplicate global b")

fail-if1.gw

1 int main()
2 {
3     if (true) {}
4     if (false) {} else {}
5     if (42) {} /* Error: non-bool predicate */
6 }
7
8 Fatal error: exception Failure("expected Boolean expression in 42")

fail-if2.gw

1 int main()

```

```

2 {
3   if (true) {
4     42;
5   } else {
6     bar; /* Error: undeclared variable */
7   }
8 }
9
10 Fatal error: exception Failure("undeclared identifier bar")

fail-print1.gw

1 int main(){
2
3   int i;
4   string s;
5
6   s="hello";
7
8   /* error, cannot print a string */
9   printi(s);
10
11  return 0;
12 }
13
14 Fatal error: exception Failure("Assign error in semant illegal argument
    found string expected int in s")

fail-return.gw

1 int main()
2 {
3   int i;
4
5   i = 15;
6   return i;
7   i = 32; /* Error: code after a return */
8 }
9
10 Fatal error: exception Failure("nothing may follow a return")

fail-struct1.gw

1 int main(){
2
3   Trackpoint t;
4   t=("hello", 3.5, 60);
5
6   return 0;
7 }
8

```



```
9 Fatal error: exception Failure("expected numeric trackpoint args")
```

```
fail-struct2.gw
```

```
1 int main(){
2
3   Trackpoint t;
4   int i;
5   t=(5.3, 6.49210, 5824);
6
7   i=getLat(T);
8
9   return 0;
10 }
```

```
12 Fatal error: exception Failure("undeclared identifier T")
```

```
fail-var1.gw
```

```
1 int main(){
2
3   string s_id;
4
5   s="hi";
6   /* invalid indentifier. cannot contain _ */
7
8   return 0;
9 }
```

```
11 Fatal error: exception Failure("lexing: empty token")
```

```
fail-while1.gw
```

```
1 int main()
2 {
3   int i;
4
5   while (true) {
6     i = i + 1;
7   }
8
9   while (42) { /* Should be boolean */
10    i = i + 1;
11  }
12
13 }
```

```
15 Fatal error: exception Failure("expected Boolean expression in 42")
```

10.15.2 Success Test Files

test-add.gw

```
1 int add(int x, int y) {
2     return x + y;
3 }
4
5 int main() {
6     printi(add(17, 25));
7     return 0;
8 }
9
10 Output:
11 42
```

test-arith1.gw

```
1 int main() {
2     printi(39 + 3);
3     return 0;
4 }
5
6 Output:
7 42
```

test-arith2.gw

```
1 int main() {
2     printi(1 + 2 * 3 + 4);
3     return 0;
4 }
5
6 Output:
7 11
```

test-arith3.gw

```
1 int main() {
2     int a;
3     a = 42;
4     a = a + 5;
5     printi(a);
6     return 0;
7 }
8
9 Output:
10 47
```

test-arith4.gw

```
1 int main() {
```

```

2     double y;
3     double z;
4
5     y = 2.3;
6     z = 2.3;
7
8     printf("y+z:");
9 }

```

10
11 Output:
12 4.6

test-array1.gw

```

1 int main() {
2     Trackpoint[] a;
3     Trackpoint b;
4     Trackpoint c;
5     Trackpoint d;
6     Trackpoint e;
7     int time;
8     int i;
9
10    b = (1.0, 2.0, 3);
11    c = (1.5, 2.9, 8);
12    d = (2.8, 89.1, 87);
13
14    a = [b];
15    a +<c>;
16    a +<d>;
17
18    for (i = 0; i < 3; i = i + 1) {
19        e = a-<i>;
20        time = getTime(e);
21        printi(time);
22    }
23 }

```

24
25 Output:
26 3
27 8
28 87

test-array2.gw

```

1 int main() {
2     int[] a;
3     int b;
4     int c;
5     int d;

```

```

6     int e;
7     int i;
8
9
10    a = [1, 2, 3, 4, 5];
11    b = 6;
12    c = 7;
13    d = 8;
14    a +<b>;
15    a +<c>;
16    a +<d>;
17
18    for (i = 0; i < 3; i = i + 1) {
19        e = a-<i>;
20        printi(e);
21    }
22 }
23
24 Output:
25 1
26 2
27 3

```

test-array3.gw

```

1  int main() {
2      int[] a;
3      int[] b;
4      int i;
5      int e;
6
7
8      a = [2, 3, 4, 5];
9      b = [1];
10
11     for (i = 0; i < 4; i = i + 1) {
12         e = a-<i>;
13         b +<e>;
14     }
15
16
17     for (i = 0; i < 4; i = i + 1) {
18         e = b-<i>;
19         printi(e);
20     }
21 }
22
23 Output:
24 1
25 2

```

26 3
27 4

test-double.gw

```
1 int main() {  
2     double a;  
3     double b;  
4     double c;  
5     a = 3.14159;  
6     b = -2.71;  
7     c = a + b;  
8     printf(c);  
9     return 0;  
10 }
```

11
12 Output:
13 0.43159

test-double1.gw

```
1 int main() {  
2     double a;  
3     a = 3.14159;  
4     printf(a);  
5     return 0;  
6 }
```

7
8 Output:
9 3.14159

test-double2.gw

```
1 int testfloat(double a, double b) {  
2     printf(a + b);  
3     printf(a - b);  
4     printf(a * b);  
5     printf(a / b);  
6     printf(a == b);  
7     printf(a == a);  
8     printf(a != b);  
9     printf(a != a);  
10    printf(a > b);  
11    printf(a >= b);  
12    printf(a < b);  
13    printf(a <= b);  
14    return 0;  
15 }  
16  
17 int main()  
18 {
```

```
19 double c;
20 double d;
21 int a;
22
23 c = 42.0;
24 d = 3.14159;
25
26 testfloat(c, d);
27
28 testfloat(d, d);
29
30 return 0;
31 }
```

32
33 Output:

34 45.1416

35 38.8584

36 131.947

37 13.369

38 0

39 1

40 1

41 0

42 1

43 1

44 0

45 0

46 6.28318

47 0

48 9.86959

49 1

50 1

51 1

52 0

53 0

54 0

55 1

56 0

57 1

test-func.gw

```
1 int add(int a, int b)
2 {
3     return a + b;
4 }
5
6 int main()
7 {
8     int a;
```

```
9   a = add(34,35);
10  printi(a);
11  return 0;
12 }
```

```
13
14 Output:
15 69
```

test-func2.gw

```
1  int printthreenum(int a, int b, int c)
2  {
3    printi(a);
4    printi(b);
5    printi(c);
6    return 0;
7  }
```

```
8
9  int main(){
10  printthreenum(420,69,100001);
11  return 0;
12 }
```

```
13
14 Output:
15 420
16 69
17 100001
```

test-func3.gw

```
1  string giveitback(string s){
2    return s;
3  }
4
5  int main(){
6    string a;
7    a = giveitback("STRAVA IS LIFE");
8    prints(a);
9    return 0;
10 }
```

```
11
12 Output:
13 STRAVA IS LIFE
```

test-func4.gw

```
1  int foo(int a) {
2    return a;
3  }
4
5  int main() {
```

```

6     return 0;
7 }
8
9 Output:
    test-func5.gw
1 int foo() {
2     return 0;
3 }
4
5 int bar(int a, bool b, int c) {
6     return a + c;
7 }
8
9 int main() {
10    printi(bar(17, false, 25));
11    return 0;
12 }
13
14 Output:
15 42

```

```

    test-func6.gw
1 int a;
2
3 int foo(int c)
4 {
5     a = c + 42;
6     return a;
7 }
8
9 int main() {
10    int a;
11    a = foo(73);
12    printi(a);
13    return 0;
14 }
15
16 Output:
17 115

```

```

    test-gcd1.gw
1 int gcd(int a, int b) {
2     while (a != b) {
3         if (a > b) {
4             a = a - b;
5         }
6         else {

```



```

7         b = b - a;
8     }
9 }
10 return a;
11 }
12
13 int main()
14 {
15     printi(gcd(2,14));
16     printi(gcd(3,15));
17     printi(gcd(99,121));
18     return 0;
19 }
20
21 Output:
22 2
23 3
24 11

```

test-gcd2.gw

```

1 int gcd(int a, int b) {
2     while (a != b) {
3         if (a > b){
4             a = a - b;
5         }
6         else {
7             b = b - a;
8         }
9     }
10    return a;
11 }
12
13 int main() {
14     printi(gcd(14,21));
15     printi(gcd(8,36));
16     printi(gcd(99,121));
17     return 0;
18 }
19
20 Output:
21 7
22 4
23 11

```

test-global1.gw

```

1 int a;
2 int b;
3

```

```

4 int printa() {
5     printi(a);
6     return 0;
7 }
8
9 int printbb() {
10    printi(b);
11    return 0;
12 }
13
14 int incab() {
15     a = a + 1;
16     b = b + 1;
17     return 0;
18 }
19
20 int main() {
21     a = 42;
22     b = 21;
23     printa();
24     printbb();
25     incab();
26     printa();
27     printbb();
28     return 0;
29 }

```

31 Output:

```

32 42
33 21
34 43
35 22

```

test-global2.gw

```

1 bool i;
2
3 int main() {
4     int i; /* Should hide the global i */
5
6     i = 42;
7     printi(i + i);
8     return 0;
9 }

```

10 Output:

```

11 84

```

test-global3.gw

```

1 int i;
2 bool b;
3 int j;
4
5 int main() {
6     i = 42;
7     j = 10;
8     printi(i + j);
9     return 0;
10 }

```

```

11
12 Output:
13 52

```

test-gpx.gw

```

1 int main(){
2     int i;
3     prints("testing gpx parser...");
4     i = numPt("test.gpx");
5     printi(i);
6     return 0;
7 }

```

```

8
9 Output:
10 testing gpx parser...
11 2345

```

test-gpx1.gw

```

1 int main(){
2     file f;
3     int time;
4     double distance;
5     double speed;
6
7     f = parseGPX("test.gpx");
8     printFirst(f);
9
10    time = totTime(f);
11    prints("total time (minutes):");
12    printi(time);
13
14    distance = totDist(f);
15    prints("total distance (km):");
16    printd(distance);
17
18    speed = avgSpeed(f);
19    prints("average speed (km/h):");
20    printd(speed);

```

```

21     stat(f);
22
23
24     return 0;
25 }
26
27 Output:
28 0.000000, 0.000000, 2345
29 total time (minutes):
30 107
31 total distance (km):
32 34.9745
33 average speed (km/h):
34 19.6118
35 ***** GPX FILE SUMMARY *****
36 Distance Covered: 34.974461 km
37 Elapsed Time: 107 min
38 Average Speed: 19.611847 km/h
39 *****

test-gpx2.gw

1 int main(){
2
3     file f;
4
5     f = parseGPX("test_short.gpx");
6     stat(f);
7
8     return 0;
9 }
10
11 Output:
12 ***** GPX FILE SUMMARY *****
13 Distance Covered: 0.020491 km
14 Elapsed Time: 0 min
15 Average Speed: inf km/h
16 *****

test-gpx3.gw

1 int main(){
2     file f;
3     f = parseGPX("test.gpx");
4     stat(f);
5     return 0;
6 }
7
8 Output:
9 ***** GPX FILE SUMMARY *****

```

```
10 Distance Covered: 34.974461 km
11 Elapsed Time: 107 min
12 Average Speed: 19.611847 km/h
13 *****
```

test-helloworld.gw

```
1 int main()
2 {
3     printi(14);
4     prints("hello world");
5     printd(5.3);
6     printb(true);
7
8
9     return 0;
10 }
```

11 Output:

```
12 14
13 hello world
14 5.3
15 1
16
```

test-if1.gw

```
1 int main() {
2     if (true) {
3         printi(42);
4     }
5     printi(17);
6     return 0;
7 }
```

8 Output:

```
9 42
10 17
11
```

test-if2.gw

```
1 int main() {
2     int x;
3     x = 2;
4
5     if (x==1) {
6         printi(42);
7     }
8     else {
9         printi(8);
10    }
11    printi(17);
```

```
12     return 0;
13 }
14
15 Output:
16 8
17 17
```

test-if3.gw

```
1 int main() {
2     if (false) {
3         printi(42);
4     }
5     printi(17);
6     return 0;
7 }
8
9 Output:
10 17
```

test-if4.gw

```
1 int main() {
2     if (false) {
3         printi(42);
4     }
5     else {
6         printi(8);
7     }
8     printi(17);
9     return 0;
10 }
11
12 Output:
13 8
14 17
```

test-if5.gw

```
1 int cond(bool b) {
2     int x;
3     if (b) {
4         x = 42;
5     }
6     else {
7         x = 17;
8     }
9     return x;
10 }
11
12 int main() {
```

```
13  printi(cond(true));
14  printi(cond(false));
15  return 0;
16  }
17
18  Output:
19  42
20  17
```

test-if6.gw

```
1  int cond(bool b) {
2    int x;
3    x = 10;
4    if (b) {
5      if (x == 10) {
6        x = 42;
7      }
8    }
9    else {
10     x = 17;
11   }
12   return x;
13 }
14
15 int main() {
16   printi(cond(true));
17   printi(cond(false));
18   return 0;
19 }
20
21 Output:
22 42
23 17
```

test-locall.gw

```
1 void foo(bool i) {
2   int i; /* Should hide the formal i */
3
4   i = 42;
5   printi(i + i);
6 }
7
8 int main() {
9   foo(true);
10  return 0;
11 }
12
13 Output:
```

14 84

test-local2.gw

```
1 int foo(int a, bool b) {
2     int c;
3     bool d;
4
5     c = a;
6
7     return c + 10;
8 }
9
10 int main() {
11     printi(foo(37, false));
12     return 0;
13 }
```

14
15 Output:

16 47

test-math.gw

```
1 int main(){
2     double y;
3     double x;
4     double d;
5
6     d = 3.14;
7     x = 25.0;
8     y = 17.0;
9
10    printi(intFloor(d));
11    printd(squareRoot(x));
12    printd(squareRoot(y));
13    printd(doubleFloor(d));
14
15    return 0;
16 }
```

17
18 Output:

19 3

20 5

21 4.12311

22 3

test-neg.gw

```
1 int main() {
2
3     int i;
```



```

4 double d;
5
6 i=3;
7 d=-6.5;
8
9 printi(i);
10 printd(d);
11
12 i=-i;
13 printi(i);
14
15 d=-d;
16 printd(d);
17
18 return 0;
19 }
20
21 Output:
22 3
23 -6.5
24 -3
25 6.5

```

test-opsl.gw

```

1 int main() {
2     printi(1 + 2);
3     printi(1 - 2);
4     printi(1 * 2);
5     printi(100 / 2);
6     printi(99);
7     printb(1 == 2);
8     printb(1 == 1);
9     printi(99);
10    printb(1 != 2);
11    printb(1 != 1);
12    printi(99);
13    printb(1 < 2);
14    printb(2 < 1);
15    printi(99);
16    printb(1 <= 2);
17    printb(1 <= 1);
18    printb(2 <= 1);
19    printi(99);
20    printb(1 > 2);
21    printb(2 > 1);
22    printi(99);
23    printb(1 >= 2);
24    printb(1 >= 1);
25    printb(2 >= 1);

```

```
26     return 0;
27 }
28
29 Output:
30 3
31 -1
32 2
33 50
34 99
35 0
36 1
37 99
38 1
39 0
40 99
41 1
42 0
43 99
44 1
45 1
46 0
47 99
48 0
49 1
50 99
51 0
52 1
53 1
```

test-ops2.gw

```
1 int main() {
2     printb(true);
3     printb(false);
4     printb(true && true);
5     printb(true && false);
6     printb(false && true);
7     printb(false && false);
8     printb(true || true);
9     printb(true || false);
10    printb(false || true);
11    printb(false || false);
12    printb(!false);
13    printb(!true);
14    printi(-10);
15    printi(--42);
16 }
17
18 Output:
19 1
```

```
20 0
21 1
22 0
23 0
24 0
25 1
26 1
27 1
28 0
29 1
30 0
31 -10
32 42
```

test-string1.gw

```
1 int main(){
2
3 string s;
4 s="hello";
5 prints(s);
6
7 s="bye";
8 prints(s);
9
10 return 0;
11 }
12
13 Output:
14 hello
15 bye
```

test-struct.gw

```
1 int main() {
2
3     Trackpoint t1;
4     Trackpoint t2;
5     double latitude;
6
7     t1=(5.5, 8.9, 32);
8     latitude = 6.4;
9     t2=(latitude, 3.5, 40);
10
11     if(getTime(t2) > getTime(t1)){
12         prints("t2 time greater than t1 time");
13     }
14
15     return 0;
16 }
```

17
18 Output:
19 t2 time greater than t1 time

test-struct1.gw

```
1 int main() {
2
3     Trackpoint t1;
4     int i;
5
6     t1=(3.5, 4.2, 593);
7     i=getTime(t1);
8     printi(i);
9
10    t1=(5.56, 25.6, 69);
11    printi(getTime(t1));
12    return 0;
13 }
```

14
15 Output:
16 593
17 69

test-var1.gw

```
1 int main() {
2     int a;
3     a = 42;
4     printi(a);
5     return 0;
6 }
```

7
8 Output:
9 42

test-var2.gw

```
1 int a;
2
3 int foo(int c) {
4     a = c + 42;
5     return 0;
6 }
7
8 int main() {
9     foo(73);
10    printi(a);
11    return 0;
12 }
13
```

14 Output:

15 115

test-while1.gw

```
1 int main() {
2     int i;
3     i = 5;
4     while (i > 0) {
5         printi(i);
6         i = i - 1;
7     }
8     printi(42);
9     return 0;
10 }
```

12 Output:

13 5

14 4

15 3

16 2

17 1

18 42

test-while2.gw

```
1 int foo(int a) {
2     int j;
3     j = 0;
4     while (a > 0) {
5         j = j + 2;
6         a = a - 1;
7     }
8     return j;
9 }
```

```
11 int main() {
12     printi(foo(7));
13     return 0;
14 }
```

16 Output:

17 14