

# COMS W4115

## Programming Languages and Translators

### Homework Assignment 2

Prof. Stephen A. Edwards    Due April 24th, 2003  
Columbia University        at 11:00 AM EST (beginning of class)

On-campus students: submit solution on paper (no email).

CVN students: Mail a PDF file to sedwards@cs.columbia.edu

Write both your name and your Columbia ID (e.g., se2007) on your solutions.

Do this assignment alone. You may consult the instructor and the TAs, but not other students.

1. Consider the following Prolog program.

```
takes(jane_doe, his201).
takes(jane_doe, cs254).
takes(ajit_chandra, art302).
takes(ajit_chandra, cs254).
classmates(X,Y) :- takes(X,Z), takes(Y,Z).
```

What does the query `classmates(jane_doe,X)` return? Give details of how the search procedure produces this result.

2. Consider the following C-like program.

```
int w = 3;
int x = 10;

void foo(y, z){
    printf("%d\n", y + y);
    x = 1;
    printf("%d\n", z);
}

int main() {
    foo(++w, ++x);
    return 0;
}
```

What does it print if the language uses

- (a) Normal-order evaluation with call-by-name?
- (b) Applicative-order evaluation with call-by-value?

3. In an assembly-language-like notation (e.g., use MIPS or a pseudocode of your own choosing), write what a good optimizing compiler would produce for the following two switch statements:

```
switch (a) {
case 1: x = 3; break;
case 2: x = 5; break;
case 3: x = 15; break;
case 4: x = 20; break;
default: x = 25; break;
}
```

```
switch (b) {
case 1: x = 3; break;
case 10: x = 5; break;
case 100: x = 15; break;
case 1000: x = 20; break;
default: x = 25; break;
}
```

4. For a 32-bit little-endian processor with the usual alignment rules, show the memory layout and size in bytes of the following C types.

```
union {
    struct {
        int a; /* 32-bit */
        char b; /* 8-bit */
    } s;
    int c;
} ul;
```

```
struct {
    char a;
    short b;
    int c;
    char d;
} sl;
```

```
struct {
    char a;
    char d;
    short b;
    int c;
} s2;
```