

# COMS W4115 Programming Languages and Translators

## Homework Assignment 3 Solutions

Prof. Ronghui Gu     Due Monday, April 22nd, 2019 at 11:59 PM

Name:

Uni:

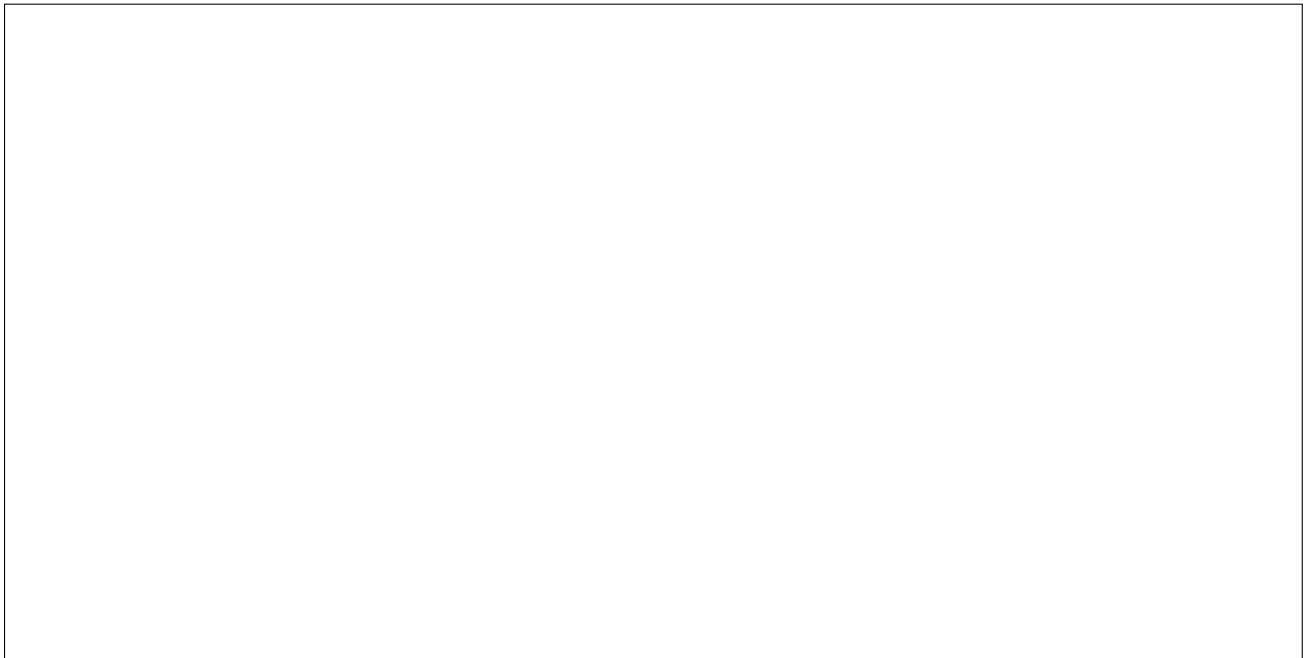
1. (40 pts.) For the following C code,

```
if (a > b) { a = a - b; } else { b = a - (c * d); }
```

(a) Construct its abstract syntax tree.

(b) Translate it in to Three-Address Code representation.

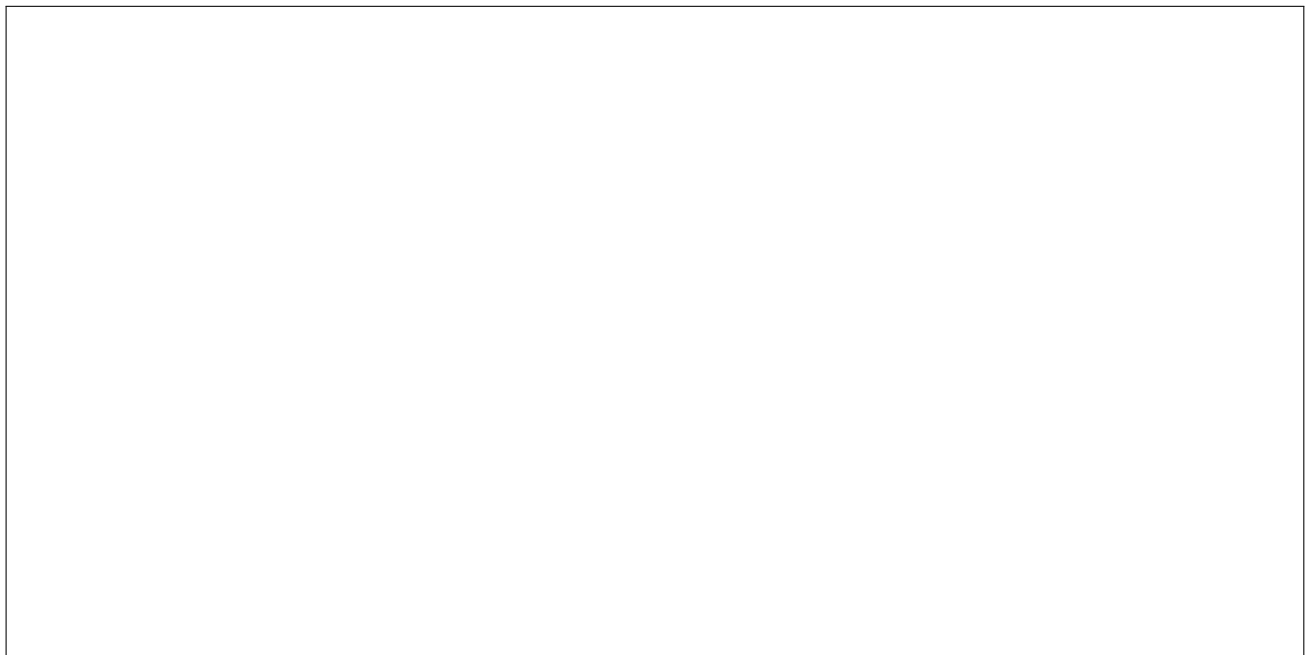
- (c) Partition the above Three-Address Code into basic blocks. Draw lines directly in the above box.
- (d) Draw the control flow graph of the above Three-Address Code.



2. (20 pts.) For the following C code,

```
[1] int x;  
[2] int main () {  
[3]     int a = 10;  
[4]     int b = 0; {  
[5]         int b = 20;  
[6]         while (a < b) {  
[7]             int x;  
[8]             x = a + 1;  
[9]             b = x;  
            ...
```

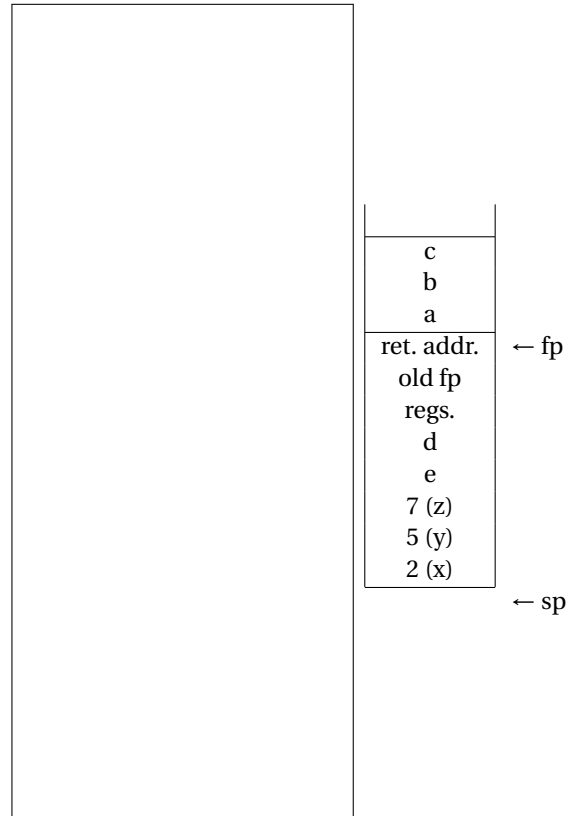
Draw the symbol table at line 9.



3. (20 pts.) Draw the layout of the stack just before *bar* is called in *foo*. Indicate storage for function arguments, local variables, return addresses, and stored frame pointers. Indicate where the stack and frame pointers point.

```
void bar(int x, int y, int z);

void foo(int a, int b) {
    int d, e;
    bar(2, 5, 7);
}
```



4. (20 pts.) For the program below written in a C-like language with nested function definitions,

```
void main() {
    int x = 5;

    void bar() {
        x = x + 2;
    }

    void foo() {
        int x = 8;
        bar();
        printf("%d\n", x);
    }

    foo(); /* Body of main() */
}
```

What would it print if the language used **static scoping**?

What would it print if the language used **dynamic scoping**?

Static scoping: 8    *bar* doesn't change local *x*  
 Dynamic scoping: 10    *bar* changes 8 into 10