

# CS314 Fall 2018

## Assignment 4

### 1 Syntax Directed Translation

Modify the recursive descent parser in Homework 3 (sample solution is posted in Sakai) such that it prints the number of assignment statements in the input program after it successfully parsed the program, and the number of addition operations. For the program listed below, your parser should print “3 assignments, 4 addition operations”.

```
def g(a, b):  
    a = 2 + b  
    if a <= b + 1: b = a + 1  
    else: b = a + 1  
    return b
```

### 2 Problem — Pointers

Given the following correct program in C,

- (a) give the correct type definitions for pointer variables `ra`, `rb`, `rc`, `rra`, `rrb`, and `rrc`.
- (b) draw a picture that shows all of the variables and their contents similar to the diagram in lecture 10. Your picture should show the variables and their values just before the first print statement (\*).
- (c) show the output from this program.
- (d) write a statement involving a pointer expression using the variables in this program which is ILLEGAL given your declared types.

```
main() {int a, b, c;  
        ??? ra; ??? rb; ??? rc; ??? rra; ??? rrb; ??? rrc;
```

```

    a = 1; b = 2; c = 3;
    ra = &a;
    rb = &b;
    rc = &c;
    ra = rb;
    rra = &rb;
    rc = *rra;
    rrc = rra;
    rc = &a;
    rrb = &rc;
    rb = &c;
    *ra = 4;
    *rb = *ra + 5;
    /* (*) */
printf ("%d %d %d\n",a,b,c);
printf ("%d %d\n",*ra,*rb);
printf ("%d %d %d\n",**rra,**rrb,**rrc);
}

```

### 3 Problem — Freeing Memory

Here is a code fragment from our singly-linked list example from class.

```

/* DEALLOCATE LIST */
for (current_cell = head;
    current_cell != NULL;
    current_cell = current_cell->next)
    free(current_cell);

```

How can you rewrite this code to make it safe? You can introduce new variables, if needed.