Exam 1

March 4th, 2015

Work alone. Do not use any notes or books. You have approximately 75 minutes to complete this exam.

Please write your answers on the exam. More paper is available if you need it. Please put your name at the top of the first page.

There are four questions on this exam, for a total of 80 points.

1 An array (20 points)

What output does the following program produce? Please draw a rectangle around your solution to distinguish it from any notes you used to produce it.

```
#include <stdio.h>
#define N (6)

int main(int argc, char **argv)
{
    int i;
    int a[N];

    for(i = 0; i < N; i++) { a[i] = i; }
    for(i = 0; i < N; i++) { a[N-i-1] = a[i]; }
    for(i = 0; i < N; i++) { printf("%d\n", a[i]); }
    return 0;
}
```

Solution

```
0
1
2
2
1
0
```

2 Every other element (20 points)

Write a function that adds one to every other element in a singly-linked list holding `int`s, starting with the first element. For example, if the list holds $1, 3, 5, 7, 9$ before calling the function, then after calling the function the list should hold $2, 3, 6, 7, 10$. The linked list type and function are declared as follows. You should assume that the last element in the list has its `next` pointer equal to 0 and that the function argument is a pointer to the first element in the list.

```
struct elt { struct elt *next; int value; };
void incrementEveryOther(struct elt *first);
```

Solution

```
#include "everyOther.h"

void
incrementEveryOther(struct elt *first)
{
    while(first) {
        first->value++;
        first = first->next;
        if(first) {
            first = first->next;
        }
    }
}
```

3 Another level of indirection (20 points)

What output does the following program produce? Please draw a rectangle around your solution to distinguish it from any notes you used to produce it.

```
#include <stdio.h>
int f(int *a, int **b) { return *(b[0] = a) + 1; }
int main(int argc, char **argv) {
    int a = 1;   int b = 2;    int c = 3;
    int *x = &a; int *y = &b;

    *x = f(&c, &y);
    printf("%d %d %d\n", a, b, c);
    *y = f(x,  &x);
    printf("%d %d %d\n", a, b, c);
    *x = f(&b, &y);
    printf("%d %d %d\n", a, b, c);
    return 0;
}
```

Solution

```
4 2 3
4 2 5
3 2 5
```

4 Bad countdown (20 points)

The following program is supposed to print the numbers from 10 down to 0, inclusive, to `stdout`. But if you compile and run it, something else will happen. What goes wrong, and how can you fix it?

```
#include <stdio.h>

int main(int argc, char **argv)
{
    unsigned int i;
    for(i = 10; i >= 0; i--) { printf("%d\n", i); }
    return 0;
}
```

Solution

The problem is that `i` is declared as an `unsigned int`, so the test `i >= 0` always succeeds, giving an infinite loop. The easiest solution is to change the declaration of `i` to just `int i`.

Curiously, the fact that `printf` uses `%d` to print this `unsigned int` will not have any noticeable effect: over the range 10 down to 0, both signed and unsigned `int`s have the same representation. But this will also be fixed by changing `i` to an `int`.