

Homework 8

Due: 2:30 PM May. 1, 2008

For this assignment, you should save all your code in the directory

```
/u/<login>/course/cs004/hw8
```

To run the handin script, enter the following command in a terminal:

```
cs004_handin hw8
```

No stencil code is provided for this homework. You should refer to the lectures and your lab assignments to get you started if you do not remember exactly how to structure a program in C.

Problem 8.1

For this question, the TAs have written some code to read in up to 10 numbers from the command line (ended by a negative number) and then print the numbers in reverse, (similar to a program we asked you to write on a previous homework). Although the code compiles without errors (it does have warnings that must be fixed), it doesn't work the way we expected it to! Look at the code on the next page, identify the bugs and turn in a corrected version of the code in a file called `reverse.c`. In comments at the bottom of the file, describe what the bugs were and what you did to fix them.

```
#include <stdio.h>

int main() {
    int array[10];
    int i, len = 0;

    printf("Enter up to 10 numbers, ending with a negative number: ");
    while(len < 10) {
        scanf("%d", array[len]);
        len++;
        if(array[len] < 0) {
            break;
        }
    }

    printf("\nReverse: ");
    for(i = len; i > 0; i--) {
        printf("%d ", array[i]);
    }
    printf("\n");
    return 0;
}
```

You can get a copy of this code by going to your `hw8` directory and then using the command

```
cp /course/cs004/pub/hw8/reverse.c .
```

Hint: Look at the code and run through it "by hand". Write down what each variable is set to and what is happening at each step in the execution of the program. After you fix the warning(s) that exist, this method should help you find the other mistake(s) in the code.

Problem 8.2

Write a program that prompts the user to enter a series of up to 15 numbers positive numbers, using a negative number to signify the end of the input. Store these numbers in an array of floats. Print out the sum and average of the numbers. Then sort the numbers using the bubble sort algorithm

(described later) and print out the sorted array. A sample run of your program should look something like this:

```
> ./bubble
Enter up to 15 positive numbers. Enter a negative number to
signify the end of input:
77.4 12 0 33.3 28 57 4.5 -1

Sum: 212.199997
Average: 30.314285
Sorted Numbers: 0.000000 4.500000 12.000000 28.000000
33.299999 57.000000 77.400002
```

Your code must include and use the following three functions:

(Your code may have other functions if you want but it MUST have these functions. These functions should be prototyped exactly in the form below. Don't change how these functions were prototyped for you by the TAs.)

```
float sum(float array[], int len);
// Outputs: the sum of the elements in the array

float average(float array[], int len);
// Outputs: the average of the elements in the array

void sort(float array[], int len);
// Sorts the elements in the array using the bubble sort algorithm
and then prints them.
```

The Bubble Sort Algorithm

In a bubble sort, you make a series of passes through your data. On each pass, you compare each element with the one following it. If the first element is greater than the element following it, swap the two elements. Otherwise, do nothing. So, if you were making a pass through 6, 4, 7, 3, 5, you would compare 6 and 4 and swap, compare 6 and 7 and do nothing, compare 7 and 3 and swap, and finally compare 7 and 5 and swap. Visually:

```
6 4 7 3 5
```

```
  ^ ^
4 6 7 3 5
  ^ ^
4 6 7 3 5
    ^ ^
4 6 3 7 5
      ^ ^
4 6 3 5 7
```

One property of this “bubbling” is that the first pass will put the largest element in position, the second will put the next-largest in position, and so on. You should repeat this process as many times as there are elements in the array, but you shouldn’t try to swap every element after the first pass through, since the numbers at the end of the array will already be sorted.

Extra Credit: Design your sort function to use recursion and earn up to 5 extra points.