CS250 Class Exercise

Date: Monday, March 9, 2009

Write the solution to as many of these exercises as you can. Make sure that you use constants and constant pointers whenever you should. As you work on these exercises take turns typing the code for each of the exercises.

Exercise 1 (num is typing):

For this exercise you will need to write the code for (or explain why its not possible) three functions. The function prototypes and the main code that will call the functions are given below. Write the functions, if possible, so the code behaves as described by the comments in main.

```
#include <iostream>

using namespace std;

void swap1(float &, float &); // swaps two floating point values
void swap2(float, float);     // swaps two floating point values
void swap3(float *,float *);  // swaps two floating point values

int main()
{
  float value1 = 2.7183;
  float value2 = 3.1416;
  cout << value1 << "," << value2 << endl; // prints 2.7183,3.1416
  swap1(value1,value2);
  cout << value1 << "," << value2 << endl; // prints 3.1416,2.7183
  swap2(value1,value2);
  cout << value1 << "," << value2 << endl; // prints 2.7183,3.1416
  swap3(&value1,&value2);
  cout << value1 << "," << value2 << endl; // prints 3.1416,2.7183
}
```

Turn in the code to Turing (01SwapPuNets). For any functions that can't be written put a comment block in the code explaining why the function can't be written.

Exercise 2 (pNum is typing):

Write a program that will use the two function prototypes shown below:

```
int *maxTwo(int *pOne, int *pTwo);
int *maxThree(int *pOne, int *pTwo, int *pThree);
```

The **maxTwo** function has two pointers to integers as parameters and returns the pointer that points to the larger integer.

The **maxThree** function has three pointers to integers as parameters and returns the pointer that points to the largest integer. The **maxThree** function should make use of the **maxTwo** function in computing its answer.

In main, ask the user to enter three values, then increment the largest value, and print all three values back to the user.

Here is sample output:

```
Enter integer values for x, y, and z: 3 8 5
The values after incrementing the largest value are:
x = 3, y = 9, z = 5
```

Turn in the code to Turing (02MaxPuNets). For any functions that can't be written put a comment block in the code explaining why the function can't be written.

Exercise 3 (num is typing):

Write a function **flipSign** that expects to be passed a pointer to an array of integers and an integer giving the array's size. The function **flipSign** should multiply each number in the array by -1, thus flipping the sign of the original value. Use only pointer syntax in the function (i.e., do not use array indexing syntax). Test that your function **flipSign** works correctly by writing a full program that includes a main function that calls **flipSign**.

Turn in the code to Turing (03flipSignsPuNets).

Exercise 4 (pNum is typing):

Write a function **check** that expects to be passed a pointer to a string (i.e., a pointer to type char) *and* a character. The function **check** should return 1 if the character is in the string, and return 0 if not. Test that your function check works correctly by writing a full program that includes a main function that calls **check**.

Turn in the code to Turing (04checkPuNets).

Exercise 5:

Write a program that reads in multiple words from a file and prints all the words out reversed. For example, if the file contains:

```
cat dog mouse rabbit
```

Your program should output to the screen:

```
tac god esuom tibbar
```

Your program should read the words into a char array. There can be as many as ten words in the file and the maximum number of characters per word is also ten. Make sure that you test your program with words that have more than ten characters and files that contain more than ten words.

Your program is to contain the following function prototype, and you are ***not*** to create a new array in this function:

`void reverseString(char *pString);`

Turn in the code to Turing (05reversePuNets).

Congratulations if you have completed all of the 5 exercises. Use the remainder of the time to work on your assignment.