In-Section Exercises - Week 2

CS50 — Fall 2011

Prepared by: Doug Lloyd '09

September 19, 2011

1. Declare and write a function `validTriangle()`, that takes three real numbers representing the lengths of the three sides of a triangle as its arguments, and outputs a true or false value corresponding to whether or not the three inputted sides are capable of making a triangle. Note the following:
   - A triangle may not have a side with negative length.
   - A triangle may not have a side with zero length.
   - The sum of the lengths of two of the sides of the triangle must be greater than the length of the third side.

   The function is called in this context:

```
#include <cs50.h>
#include <stdio.h>

int main(int argc, char *argv[]) {

   printf("Please input, one at a time, the lengths of the sides of a triangle.\n");

   printf("First side: ");
   int a = GetFloat();
   printf("Second side: ");
   int b = GetFloat();
   printf("Third side: ");
   int c = GetFloat();

   bool valid = validTriangle(a, b, c);

   if(valid)
      printf("This is a valid triangle!\n");
   else
      printf("This is NOT a valid triangle!\n");
}
```

2. Write a function that accepts as its parameters an integer array (called, say, `arr`) and an integer (called, say, `size`) representing the size of that array. In the body of the function, determine whether that array is in ascending numerical order from `arr[0]` to `arr[size-1]`. If it is, return the string "yes" to the calling function. Otherwise, return the string "no". Do **NOT** modify the contents of `arr`!
3. Write a program with a partner that conforms to the following specifications:
   - The program must accept 1 (and only 1) command line argument, besides the program's name.
   - If that character is in the range of [A-Z] or [a-z], the program prints, on a new line, only that character's placement in the alphabet, counting from 0 [0-25].
   - Otherwise, it prints -1.

   Additionally, you may assume the following:
   - The command line argument inputted will be a character (but, recall the type of `argv`!)