SOF103 C and C++ Programming
Lab Exercise 7
Functions – Passing Arguments & Recursive Functions

Part A: Review Questions

1. Write the function prototype for each of the following:
   (a) Function `hypotenuse` that takes two floating-point arguments `side1` and `side2`, and returns a floating-point result.
   (b) Function `smallest` that takes three integers, `x`, `y`, `z` and returns an integer.
   (c) Function `instructions` that does not receive any arguments and does not return a value.
   (d) Function `intToFloat` that takes an integer argument, `number`, and returns a floating point result.
2. Write a C++ statement that assign random integers to the variable *n* using rand() function in the following ranges:
   (a) $1 \le n \le 100$
   (b) $0 \le n \le 9$
   (c) $1000 \le n \le 1112$
   (d) $-1 \le n \le 1$
   (e) $-3 \le n \le 11$
3. Write the value of `x` after each of the following statements is executed:

(a)	`x = fabs(7.5)`	
(b)	`x = floor(7.5)`	
(c)	`x = fabs(0.0)`	
(d)	`x = ceil(0.0)`	
(e)	`x = fabs(-6.2)`	
(f)	`x = ceil(-6.3)`	
(g)	`x = ceil(-fabs(-8+floor(-5.5)))`	

4. Write the function prototype for the following:
   (a) Function `sumArray` that sums the value of an integer array `a`[10] and returns the result as an integer.
   (b) Function `countChar` that determines the number of characters in a string. The string is passed as a pointer to a string `*sPtr` and the result is return as an integer.

(c) Function `avgValue` that calculates the average of a floating-point array of type double `val`[5]. The result is passed by reference in the argument of the function.

(d) Function `printString` that prints a character string that is passed as a pointer `*stringPtr` in the argument of the function.

(e) Function `convertTemperature` that converts a temperature value in degree Celsius to Fahrenheit. The temperature values are type **`double`**. Use function **call-by-value**.

(f) Repeat exercise (e) above but use function **call-by-reference**.

Part B: Programming Practices

1. Write a program that plays the game of "guess the number" as follows: Your program chooses the number to be guessed by selecting an integer at random in the range 1 to 1000. The program then types: [Hint: use rand() and srand()]

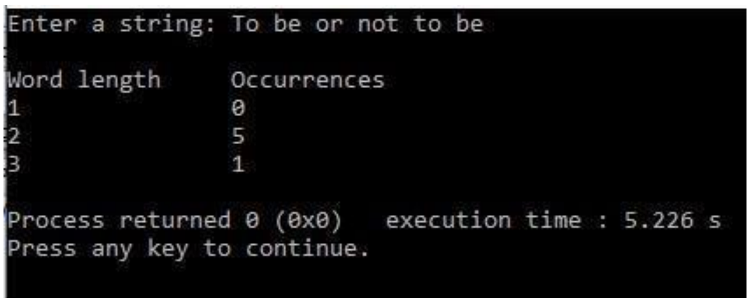
```
I have a number between 1 and 1000.
Can you guess my number?
Please type your first guess.
```

The player then types a first guess. The program responds with one of the following:

```
1. Excellent! You guessed the number!
   Would you like to play again (y or n)?
2. Too low. Try again.
3. Too high. Try again.
```

If the player's guess is incorrect, your program should loop until the player finally gets the number right. Your program should keep telling the player **`Too high`** or **`Too low`** to help the player "zero in" on the correct answer.

2. Write a program that reads a line of text and prints a table indicating the number of one-letter words, two-letter words, three-letter words, etc. appearing in the text. For example, the phrase: "To be or not to be" contains



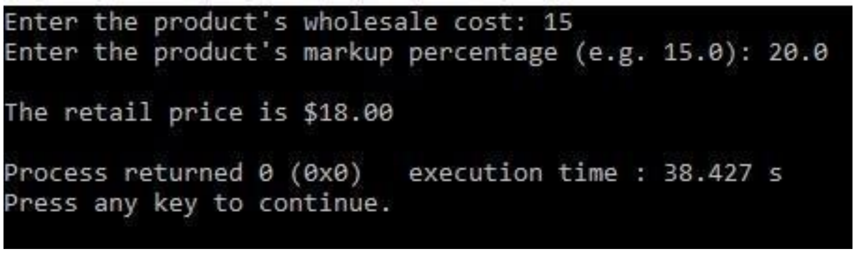
```
Enter a string: To be or not to be

Word length     Occurrences
1               0
2               5
3               1

Process returned 0 (0x0)   execution time : 5.226 s
Press any key to continue.
```

Hint: use built-in functions strtok() and strlen() defined in <cstring>

3. Write a program that takes the following input from the user:
   a. Wholesale price of the product
   b. Markup percentage

   The program should then calculate the retail price of the product by defining a user-defined function "RetailPrice" which receives the wholesale price and markup percentage as function arguments and returns the retailprice of the product.

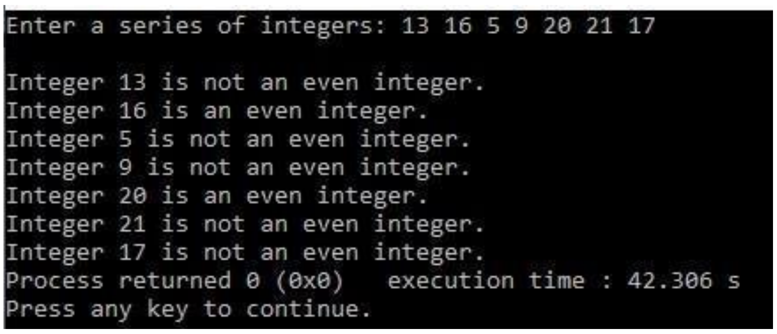
Sample program run:



```
Enter the product's wholesale cost: 15
Enter the product's markup percentage (e.g. 15.0): 20.0

The retail price is $18.00

Process returned 0 (0x0)   execution time : 38.427 s
Press any key to continue.
```

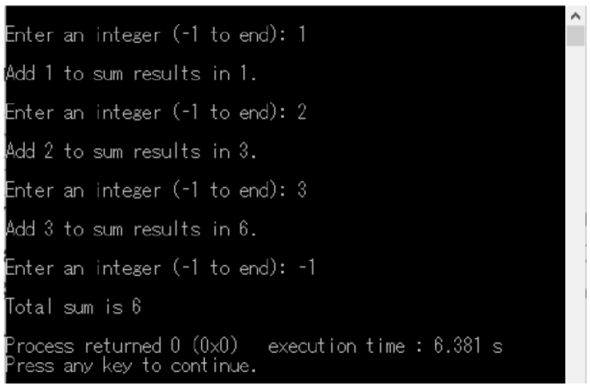
4. Write a program that inputs a series of integers and passes them one at a time to function `even` which uses the modulus operator to determine if an integer is even. The function should take an integer argument and return **`true`** if the integer is even and **`false`** otherwise.

Sample program run:



```
Enter a series of integers: 13 16 5 9 20 21 17

Integer 13 is not an even integer.
Integer 16 is an even integer.
Integer 5 is not an even integer.
Integer 9 is not an even integer.
Integer 20 is an even integer.
Integer 21 is not an even integer.
Integer 17 is not an even integer.
Process returned 0 (0x0)   execution time : 42.306 s
Press any key to continue.
```

5. Write a function `cumulativeSum` that sums an integer passed to the function with the previous accumulated `sum` value. Use pass-by-reference in the function argument for the `sum` variable. An example of program run is shown below.



```
Enter an integer (-1 to end): 1
Add 1 to sum results in 1.
Enter an integer (-1 to end): 2
Add 2 to sum results in 3.
Enter an integer (-1 to end): 3
Add 3 to sum results in 6.
Enter an integer (-1 to end): -1
Total sum is 6
Process returned 0 (0x0)   execution time : 6.381 s
Press any key to continue.
```

6. One function takes in a series of **`double`** values and then find the largest number out of the numbers and another function finds the smallest numbers out of the series of numbers. Write the two functions if:

   (a) function `largest` is called by-reference with pointer argument

   (b) function `smallest` is called by-reference with array argument

7. Write a program to calculate the statistics of student marks. There are 10 student marks stored in a floating-point array of type **`double`**: {85.0, 79.5, 81.0, 77.5, 78.0, 70.0, 72.5, 85.5, 90.5, 73.0}. Use a function to calculate the mean of the student marks based on the equation as follows:

$$\mu = \frac{\sum_{i=1}^{N} x_i}{N}$$

   Use another function to calculate the standard deviation of the marks based on the equation as follows:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N} (x - \mu)^2}{N}}$$

   Return the results of the calculation to the `main` function to be printed out.

8. Write a program that receives a string of characters from the user input. Then pass the string to a function `convertString` to convert all characters to capital letters. Return the string with all capital letters to the `main` function.

   Next, pass the string with all capital letters to another function `printString` to be printed out on screen.

9. The Fibonacci series 0, 1, 1, 2, 3, 5, 8, 13, 21 ... begins with the terms 0 and 1 and has the property that each succeeding sum of the two preceding terms. Write a recursive function `fibonacci(n)` that calculates the *n*th Fibonacci number.

10. Write a recursive function `power(base, exponent)` that when invoked returns the value of $base^{exponent}$. For example, `power(3,4) =` 3*3*3*3. Assume that exponent is an integer greater than or equal to 1. <u>Hint</u>: The recursion step would use the relationship

$$base_{exponent} = base \cdot base_{exponent - 1}$$

and terminating condition occurs when *exponent* is equal to 1 because $base^1 = base$.