

Homework1

- 1) Write a program which checks if a given point is located on the circumference of a circle defined by its center and rayon.
- 2) Write a program which gives a number n from the user and returns the factorial of n .
- 3) Write a program that reads a character c from the keyboard, and
 - (a) if c is a **lower case** letter, displays its corresponding upper case letter
 - (b) if c is a **upper case** letter, displays its corresponding lower case letter
 - (c) if c is not a letter, displays an error message indicating that the character c is not an alphabet letter
- 4) we want to depict a pyramid of height $height$ as illustrated in the figure for $height = 6$

```

      * *
    * * *
  * * * *
* * * * *
* * * * *
* * * * *

```

- (a) Write a function `print_pyramid(int height)` which takes a single integer argument `height` and displays a pyramid of this height
- (b) Test the function with a simple driver program, which should be able to reproduce the following output examples:

```

how high would you like the pyramid? 37
Pick another height (must be between 1 and 30): 6

```

- 5) Write a function `integerPower(base, exponent)` that returns the result of $base^{exponent}$. For example, `integerPower(3, 4)` returns $3^4 = 81$. Assume that `exponent` is a positive, nonzero integer and that `base` is an integer. Do not use any math library functions.
- 6) We want to create a program which calculates the average and the standard deviation of 1, 2, 3 or 4 real numbers.
 - (a) Create a header file named `statistics.h` which contains declaration of variables and functions needed by this program.
 - (b) Create an implementation file `statistics.cpp` which contains the implementation of functions `average()` and `standardDeviation()`.

The standard deviation of the numbers r_1, \dots, r_N is defined as the square root of the average of the expressions:

$$((r_1 - a) * (r_1 - a)) + \dots + ((r_N - a) * (r_N - a))$$

where a is the average value of r_1, \dots, r_N .

- (c) Write a program which for a given number of inputs introduced by the user tests the functions defined in *statistics.h*. The program should continue until the user introduce the EOF character (wants to finish the execution of the program). Your program should be able to reproduce the following sample input/output session:

Do you wish to test 1, 2, 3 or 4 numbers (enter 0 to end the program): 3

Enter first value: 5

Enter second value: 7

Enter third value: 9

Average: 7. Standard deviation: 1.63299.

Do you wish to test 1, 2, 3 or 4 numbers (enter 0 to end the program): 1

Enter first value: 5.8

Average: 5.8. Standard deviation: 0.

Do you wish to test 1, 2, 3 or 4 numbers (enter 0 to end the program): 8

Sorry, the program can only test 1, 2, 3 or 4 values.

Do you wish to test 1, 2, 3 or 4 numbers (enter 0 to end the program): 0

- 7) Write a program that asks the user to type 10 integers of an array. The program must compute the average of 10 integers then write the number of integers greater than average.
- 8) Write a program that asks the user to type 10 integers of an array. The program will then display either "the array is growing", "the array is decreasing", "the array is constant", or "the array is growing and decreasing".
- 9) An array is a palindrome if it reads the same in both directions. For example,

BOB - LEVEL - DAD - MOM - REFER

are the palindrome word.

- (a) Write a Non - recursive *IsPalindrome()* function which takes appropriate argument and check if the word is palindrome, print " It's a palindrome word", otherwise print "It's not a palindrome word".
- (b) Write a recursive *IsPalindrome()* function do the same task.

- 10) Write a program that asks the user to type 10 integers of an array and an integer value key. The program must search if the value key exists in the array and must remove the first occurrence of key, shifting each following element left and adding a zero at the end of the array. The program must then write the final array.
- 11) Write a *IsEqual(char array1 [], int array1Size , char array2 [], int array2Size)* function which takes two char array as an argument then check if they are equal or not, your program should be able to produce something like this:

Enter first array: ...

Enter second array: ...

array1 = array2

OR

array2 != array2

- 12) Write a program that asks the user to type 10 integers of an array. The program must sort the array and then write the final array.

(a) Write a function *bubbleSort(int [], int)* which takes array and the array size as an argument then sort the array.

NOTE: Bubble sort is a simple sorting algorithm that works by repeatedly stepping through the list to be sorted, comparing each pair of adjacent items and swapping them if they are in the wrong order. The pass through the list is repeated until no swaps are needed, which indicates that the list is sorted.

(b) Write a function *insertionSort(int [], int)* which takes array and the array size as an argument then sort the array.

NOTE: Insertion sort a simple, but inefficient, sorting algorithm. The first iteration of this algorithm takes the second element and, if it's less than the first element, swaps it with the first element (i.e., the program inserts the second element in front of the first element). The second iteration looks at the third element and inserts it into the correct position with respect to the first two elements, so all three elements are in order. At the i' th iteration of this algorithm, the first i elements in the original array will be sorted.

- 13) Write a *is_Symmetric* (*int* *[][size]*) function which takes a square matrix then check if it is a symmetric matrix or not.

NOTE: In linear algebra, a **symmetric matrix** is a square matrix that is equal to its transpose, for example:

for all indices *i* and *j* , the following 3×3 matrix is symmetric:

$$\begin{bmatrix} 1 & 7 & 3 \\ 7 & 4 & -5 \\ 3 & -5 & 6 \end{bmatrix}$$

- ✓ It is very important that you write easily readable, optimized, well-designed code. for all exercises please follow the Google C++ style guide
<<http://google-styleguide.googlecode.com/svn/trunk/cppguide.xml>>
- ✓ You may not use any other external libraries
- ✓ Please write and upload your codes in separate files that named by question number!

Vahid Kharazi <vahid@kharazi.net>