

CS-1002 Programming Fundamentals Fall-2022

ASSIGNMENT-04

**Submission deadline:** 16<sup>th</sup> November 2022 04 pm

Instructions:

1. Assignments are to be done individually. You must complete this assignment by yourself. You cannot work with anyone else in the class or with someone outside of the class. The code you write must be your own and you must understand each part of your code.
2. Use appropriate data types, operations, conditional structures, loops and arrays.
3. Do not use any String or math libraries (such as cmath etc) and also do not use built-in function (such as pow etc). **Caution:** zero marks will be awarded.
4. Do not edit **Function Prototypes**.
5. **Your code must be generic.** Size of arrays must be constant. If TA change sizes of array your code should not break or produce syntax or logical errors else zero marks will be awarded.
6. Marks distribution and test examples are provided with each question (Hangman game working: <https://www.hangmanwords.com/play>). Your code will be evaluated with **similar test examples**. If the required output is generated, you will be awarded full marks. Failing to generate the correct output will result in zero marks. Total Marks: 100
7. Course Late submission policy will be applied.
8. The output should be properly displayed and well presented. 10% mark will be deducted in each question if appropriate comments and indentation not done in source code.
9. **Plagiarism:** Plagiarism of any kind (copying from others, copying from the internet, etc) is not allowed. If found plagiarized, you will be awarded zero marks in the assignment. Repeating such an act can lead to strict disciplinary actions and failure in the course.
10. **Submission Guidelines:** Dear students, we will be using auto-grading tools, so failure to submit according to the below format would result in zero marks in the relevant evaluation instrument.
  - a. For each question in your assignment, if necessary, make a separate .cpp file e.g. for question 1, make ROLL-NUM\_SECTION\_Q1.cpp, and so on. Each file that you submit must contain your name, student-id, and assignment # on the top of the file in the comments.
  - b. Combine all your work in one folder. The folder must contain only .cpp files (no binaries, no exe files etc.). If we unable to download your submission due to any reason you will be awarded zero mark.
  - c. Run and test your program on a lab machine before submission. If there is a syntax error, zero marks will be awarded in that specific question.
  - d. Rename the folder as ROLL-NUM\_SECTION (e.g. 22i-0001\_A) and compress the folder as a zip file. (e.g. 22i-0001\_A.zip). Only zip file will be acceptable.
  - e. Submit the .zip file on Google Classroom within the deadline (**16<sup>th</sup> November, 2022 4 pm**).
  - f. Submission other than Google classroom (e.g. email etc.) will not be accepted.
  - g. The student is solely responsible to check the final zip files for issues like corrupt files, viruses in the file, mistakenly exe sent. If we cannot download the file from Google classroom due to any reason it will lead to zero marks in the assignment.

Note: Follow the given instruction to the letter, failing to do so will result in a zero.

(Marks 5) **Problem 01:**

**Implement the Euclidean Algorithm for finding the greatest common divisor of two given positive integers.**

The Euclidean Algorithm transforms a pair of positive integers (m, n) into a pair (d, 0) by repeatedly dividing the larger integer by the smaller integer and replacing the larger with the remainder. When the remainder is 0, the other integer in the pair will be the greatest common divisor of the original pair (and of all the intermediate pairs).

For example, if **m** is 532 and **n** is 112, then the Euclidean Algorithm reduces the pair (532,112) to (28,0) by (532,112) --> (112,84) --> (84,28) --> (28,0). So 28 is the greatest common divisor of 532 and 112.

Function prototype: int euclideanGCD(int , int);

(Marks 10) **Problem 02:**

Write and test the “digit” function:

**Function Prototype:** int digit(int n,int k)

This function returns the kth digit of the positive integer n. For example, if n is the integer 85,419, then the call digit(n,0) would return the digit 8, and the call digit(n,2) would return the digit 4.

Examples:

1. Input: n = 25419 , k = 1 output: 5
2. Input: n = 2 , k = 0 output: 2
3. Input: n = 2 , k = 1 output: index out of bound (return -1)

Note:

- The digits are numbered from left to right beginning with the “zeroth” digit.
- Take input and display output in main function.
- Do not use array or string etc.

(Marks 10) **Problem 03:**

Write a C++ function that evaluates polynomials an  $x^n + a_{n-1} x^{n-1} + \dots + a_2 x^2 + a_1 x^1 + a_0 x^0$ . It should take following inputs:

- Value of x.
- Highest power n.
- Coefficients in descending order  $a_n, a_{n-1}, \dots, a_2, a_1, a_0$  (coefficients will be n+1)

For instance, if x=4 , n=3 & coefficients are [2,3,1,2], then output should be 182 which is obtained by evaluating the polynomial  $2 * 4^3 + 3 * 4^2 + 1 * 4^1 + 2 * 4^0$

**Function Prototype:** double evaluatePolynomial();

Example01:

x = 4 , p = 3

Enter 4 coefficients in descending order of power  
2, 3, 1, 2  
output: 182

Example02:  
x = -2.5, p = 7  
Enter 8 coefficients in descending order of power  
-2, 0, 0, -4.6, 0, 1, 6, 0  
output: 1032.265625

Example03:  
x = -2.5, p = 0  
Enter 1 coefficients in descending order of power  
7  
output: 7

Example04:  
x = -2.5, p = -4  
output: Error (power must be >=0)

Note:

- Do not use built-in power function. And power must be >= 0 and integer.
- Do not use array or string etc.
- Read inputs inside evaluatePolynomial function.
- Display output in main function.
- Apply input validations

(Marks 25) **Problem04:**

In this problem, you are required to use for/while loop only to implement a calculator **using ++ (increment) and -- (decrement) operators only**. The operations that your calculator shall perform are limited, i.e. **addition, subtraction, multiplication and absolute**. Your program shall ask the user about the two input values and an operator, and then use the switch-case structure to run the required operation (addition (+), subtraction (-), multiplication (\*) and absolute (~)).

Prototype are:

- int addition(int n1,int n2);
- int subtraction(int n1,int n2);
- int multiplication(int n1,int n2);
- int absolute(int n1);

Op	n1	n2	Answer
+	10	12	22
+	10	-12	-2
+	-10	12	2
+	-10	-12	-22
-	10	12	-2

-	10	-12	22
-	-10	12	-22
-	-10	-12	2
*	10	12	120
*	10	-12	-120
*	-10	12	-120
*	-10	-12	120
~	5		5
~	-5		5

Note:

1. Any operation that uses arithmetic/bitwise operators (+, -, \*, /, %, +=, -= etc.) other than ++ or -- will be awarded zero marks.
2. All functions inputs can be positive or negative. absolute function take's only one argument.
3. Take input and display output in main function.
4. **05 Marks for Menu** (menu must display your aesthetic sense).

(Marks 20) **Problem 05: Sorting and Merging**

Write a program that takes 5 integer arrays as input. You have to ensure that user enter these arrays in ascending order, if user enters incorrectly display a prompt to “Read input in correct format” and input again. Write a C++ program to produce an array that merges elements of all arrays in descending order, but it also needs to remove duplicates.

Example 01:

Array A1: {1, 5, 6}

Array A2: {1, 2, 4, 8}

Array A3: {5, 6, 7, 8}

Array A4: {23, 24, 94, 100, 105, 115}

Array A5: {1, 2, 5, 6, 8, 24, 24, 67, 107, 108}

Merged array: {115, 108, 107, 105, 100, 94, 67, 24, 23, 8, 7, 6, 5, 4, 2, 1}

Example 02:

Array A1: {1, 3, 5, 6}

Array A2: {1, 2, 4, 8}

Array A3: {5, 6, 7, 8, 1234}

Array A4: {23, 24, 94}

Array A5: {1, 2, 2, 23, 24, 67, 108}

Merged array: {1234, 108, 94, 67, 24, 23, 8, 7, 6, 5, 4, 3, 2, 1}

Note:

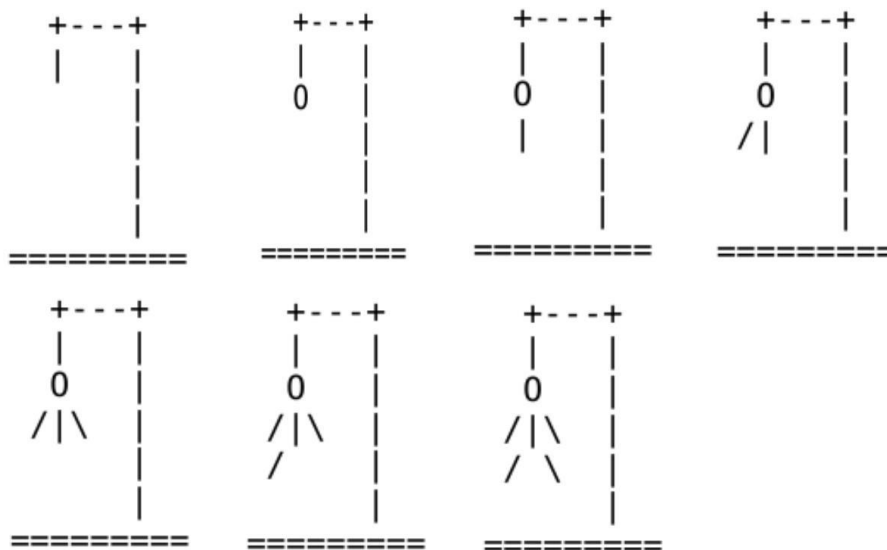
1. Display **Merged array: {1234, 108, 94, 67, 24, 23, 8, 7, 6, 5, 4, 3, 2, 1}** (Example 02).
2. Apply input validation.
3. Code must be generic: Sizes of all arrays must be constant. Remember if arrays sizes changed, your code should not break (due to segmentation fault) or produce syntax or logical errors.

(Marks 30) **Problem 06:**

**HangMan or HangWoman:** In this question you are going to write code for hangman. To write your program you will need a list of words, you can create your own list of words (single char array of 100 size).

For example, input list of Words = "fast,programming,student,are,lazy,hangmen" .

You can display following seven types of hangmen at each level of your game



Note: assume all the words are in small alphabets. And input words must be comma (,) separated.

Game link: <https://www.hangmanwords.com/play>