## NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES ISLAMABAD CAMPUS

**Object Oriented Programming- SPRING 2024 ASSIGNMENT- 1**

Due Date: Thursday 22nd February 2024 at 11:59 pm on Google Classroom

# Instructions: **Total Marks: 110**

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 your write must be your own and you must have an understanding of each part you code. You are encouraged to get help from the instructional staff through email, on google classroom or individually visiting their respective offices.
2. The AIM of this assignment is to give you practice with Pointers and Dynamic memory allocation in c++.
3. No late assignments will be accepted.
4. Displayed output should be well mannered and well presented. Use appropriate comments and indentation in your source code.
5. Plagiarism:

Plagiarism of any kind (copying from others, copying from 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 course.

# 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.

* 1. For each question in your assignment, make a separate cpp file e.g. for question 1, make q1.cpp and so on. Each file that you submit must contain your name, student-id, and assignment # on top of the file in the comments.
  2. Combine all your work in one folder. The folder must contain only .cpp files (no binaries, no exe files etc.).
  3. Run and test your program on a lab machine before submission.
  4. Rename the folder as ROLL-NUM\_SECTION (e.g. 23i-0001\_B) and compress the folder as a zip file. (e.g. 23i-0001\_B.zip).
  5. Submit the .zip file on Google Classroom within the deadline.
  6. Submission other than Google classroom (e.g. email etc.) will not be accepted.

The student is solely responsible to check the final zip files for issues like corrupt file, virus 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.

**Problem 1 (20 marks):**

Create a C++ program to simulate the **Boggle Word Search** Game. The program should encompass the following features:

Read the Boggle grid from a file named "Boggle.txt" and store it in a 2D char array followed by prompting the user to enter a word to search in the Boggle grid. Convert the user's input (string) into a char array for processing and implementing a word search function that checks for the entered word horizontally, vertically, and diagonally in the grid.

If the word is found, display the starting and ending positions of the word, else print "Word not found."

Sample content for "boggle.txt":

RETAWSTLYB

POIUCMPTAE

SEHDRIEVOF

RSNAIOLETC

AEEECARSST

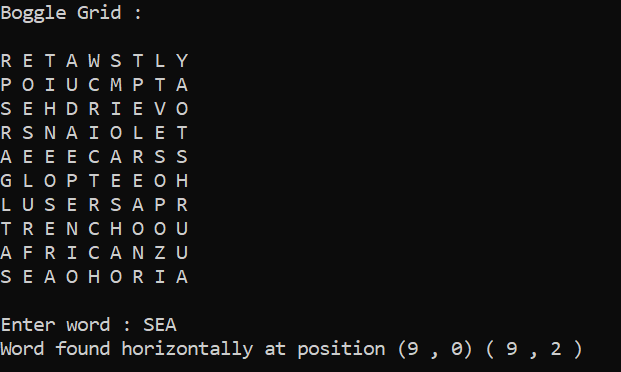
GLOPTEEOHY

LUSERSAPRO

TRENCHOOUR

AFRICANZUU

SEAOHORIAN



**Problem 2 (10 marks):**

Design and implement a 2D dynamic array in C++. The dynamic array should allow users to dynamically allocate and deallocate memory for an array of integers. Your task is to implement following functions:

1. Create and initialize an array.
2. Add elements to dynamic array.
3. Get size of array.
4. Access elements by index.
5. Resize an array dynamically.
6. Display elements of array.
7. Deallocate memory.

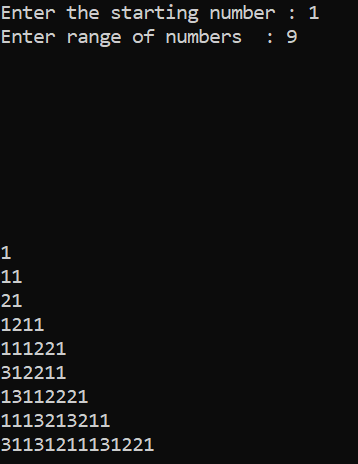
**Problem 3 (20 marks):**

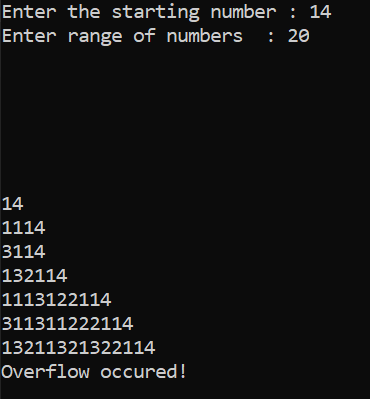
**Look-And-Say** is a sequence of numbers in which each term describes the previous one. For example:

1. 1 (read as one)
2. 11 (read as one ‘1’)
3. 21 (read as two 1's)
4. 1211 (read as one 2, one 1) and so on…

Every line describes the numbers present in the previous one.

Your task is to design a C++ program that generates a **Look-and-Say** sequence for any number **“m”** till **“n”** terms. Prompt the user to enter the starting number for the Look-and-Say sequence, and the range **‘n’**. Implement a function to generate the Look-and-Say sequence and store each term in an array. Display the generated Look-and-Say sequence in the array.

**Note:** Handle cases where the generated term exceeds the capacity of a **long long int** (2^63 -1) by displaying "overflow occurred." Also ensure that both “m” and “n” are greater than zero. If an overflow occurs, indicate it in the output along with the previously generated terms.



**Problem 4 (20 marks):**

Write a C++ program to perform various operations on 3D dynamically created matrix. The program should allow user to initialize the matrix with random values, perform arithmetic operations, find transpose of matrix and display elements. You are required to:

* Create a 3D matrix dynamically and initialize it with random values ranging from 10 to 99.
* Write a function to add scalar value to all elements of array
* Perform addition, subtraction, multiplication and division operations on matrix
* Create a transposed matrix by interchanging the rows and columns of the matrix.
* Deallocate memory for original matrix and assign transpose matrix to original matrix pointer
* Display elements of matrix in formatted manner
* Properly deallocate memory for matrix to prevent memory leaks

**Problem 5 (20 marks):**

Write a C++ program that focuses on string manipulation using character arrays. Your task is to implement various functions for string operations, including string length calculation, conversion from string to a character array, concatenation of character arrays, and a custom string comparison function. The comparison function should handle matching characters, **'#'** as a **wildcard**, and **'\*'** as a **repetition specifier**.

For the conversion function:

Given an input string **“S”** and a pattern **“P”**, implement expression matching. Modifications to the Comparison function are as follows:

* **'#'** in **“P”** matches any single character in **“S”**.
* **'\*'** in **“P”** matches zero or more instances of the preceding element in “S”

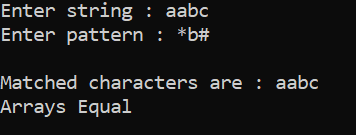
The matching should cover the entire input string. Finally display the characters that matched between the strings. Use the following **function prototypes**:

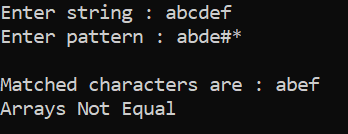
1. bool compare (const char\*, const char\*) // function to compare string and the pattern
2. char\* concat (const char\*, const char\*) // function to concatenate two char arrays
3. char\* convert (const string&) // function to convert string into char array
4. int len (const char\* arr1) // function to find length of char array

**Note**: You are required to take input as string and then implement all the other functionalities

using char arrays.

**Sample Solution:**





**Problem 6 (20 marks):**

Design a C++ program that performs **matrix pattern drawing**, encryption, decryption, and grid updates.

The program incorporates the following features:

Your task is to create a 2D character array (grid) with dimensions 5x5 and prompt the user to enter a symbol to draw the pattern with. Implement functions to display the matrix and draw either an **‘F’** or **‘E’** pattern. Implement functions to encrypt and decrypt the matrix using a **transpose and column swap** method. Prompt the user to enter a symbol and an index followed by implementing functionality to update the grid at the specific index with the symbol entered by the user (the index is updated only if it was part of the pattern created earlier). Display the initial matrix, the drawn pattern within the pattern, encrypted matrix, decrypted matrix, and the matrix after updating an index value.

Note: Implement functionality for only one of the patterns (either 'E' or 'F')

:) Draw an 'F' pattern if u feel the assignment was difficult.

:) Draw an 'E' pattern if u feel the assignment was easy.

For **“F”**



For **“E”**

