

NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES ISLAMABADCAMPUS

CS-1002 Programming Fundamentals Fall-2023

ASSIGNMENT-03

Section (A, B, C, D, E, F, G, H, J and K)

**Deadline: 10th November 2023 20:00 .**

**Instructions:**

- For each question in your assignment, make a separate cpp file e.g. for question 1, make ROLL-NUM\_SECTION\_Q#.cpp (23i-0001\_A\_Q1.cpp) and so on. Each file that you submit must contain your name, student-id, and assignment # on top of the file in comments.
- Combine all your work in one folder. The folder must contain only .cpp files (no binaries, no exe files etc.).
- Rename the folder as ROLL-NUM\_SECTION (e.g. 23i-0001\_A) and compress the folder as a zip file. (e.g. 23i-0001\_A.zip). Do not submit .rar file.
- All the submissions will be done on Google classroom within the deadline. Submissions 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 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.
- Displayed output should be well mannered and well presented. Use appropriate comments and indentation in your source code.
- **Be prepared for viva that involves modifying code after the submission of assignment.**
- If there is a syntax error in code, zero marks will be awarded in that part of assignment.
- Understanding the assignment is also part of assignment.
- **The AIM of this assignment is to give you practice with functions and arrays**
- **Zero marks will be awarded to the students involved in plagiarism. (Copying from the internet is the easiest way to get caught).**
- **Late Submission policy will be applied as described in course outline.**

**Tip: For timely completion of the assignment, start as early as possible.**

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

---

## Question 1

Complete the following programs by writing all functions.

```
1.
int main()
{
//declare identifiers
int x, y, z, max, min;
//read three integer numbers
into x,y,z in main
//find and display the sum and
average of x,y,z
int Sum; float Average;
FindSum(x,y,z);

ShowSum(Sum);
FindAvg(x,y,z)
ShowAvg(average)
//find and display the max. and
min. of x,y,z
FindMax(x,y,z);
ShowMax(max); //terminate
FindMin(x,y,z);
ShowMin(min); //terminate

program system("pause"); return
o;
}
```

### 2. Practice Arthemtic

```
int main()
{ // display menu
DisplayMenu();
//select your choice
int choice;
Select( choice); //
test your choice
switch( choice )
{ case 1: PracticeAdd(); break; case
2:
PracticeSub(); break;
}
//terminate program
system("pause"); return 0
}
```

Enter 3 integer numbers: 10 20 30

Sum = 60

Average = 20.00

Maximum = 30

Minimum = 10

-----Practice Arithmetic----- 1.

Addition

2. Subtraction

Enter your choice (1/2): 1

10 + 20 =? 30

CORRECT

Continue(y/n)? y

11 + 14 =? 35

WRONG

Continue(y/n)? n

No. of CORRECT answers = 1

No. of WRONG answers = 1

## Question 2

You are given two  $N \times N$  square matrices, A and B. Each of the elements of these matrices is an integer between 1 and K(inclusive). You have to convert matrix A into matrix B in minimum number of operations. In each operation you can choose one element of matrix A and change it to any integer between 1 and K (inclusive). You have to ensure that after any operation the matrix is not converted to a symmetric matrix. A square matrix is said to be symmetric if j-th element of i-th row is equal to the i-th element of j-th row for all (i, j) where  $1 \leq i \leq N$  and  $1 \leq j \leq N$ . For example:

$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 2 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix}$
Symmetric Matrix	Non-symmetric Matrix

Input: Input will start with an integer T ( $T \leq 200$ ), number of test cases. Each test case starts with a line containing two integers N ( $1 \leq N \leq 100$ ) and K ( $1 \leq K \leq 9$ ). This line will be followed by 2N lines. First N lines will represent matrix A and next N line will represent matrix B. Each of these 2N lines will contain N integers, all of these integers are in between 1 and K (inclusive)

Output: For each test case, output a single line containing the case number followed by the minimum number of operations required to convert A into B. If it is impossible to convert A into B obeying the rules, print '-1' instead. See output for sample input for exact formatting

Sample input:

```

3
3 9
1 2 3
4 5 6
7 8 9
1 2 3
4 5 6
7 8 9
2 3
1 2
1 1
1 1
3 1
2 3
1 2
3 1
1 3
2 1

```

Sample output:

Case 1: 0

Case 2: 2

Case 3: 3

You will be submitting testing the validity of solution to question 1 using domjudge. DOMjudge is an automated system to run programming contests, like the ICPC.

A separate guide will be shared for the testing of Q1 in coming days.

<https://www.domjudge.org/demo>

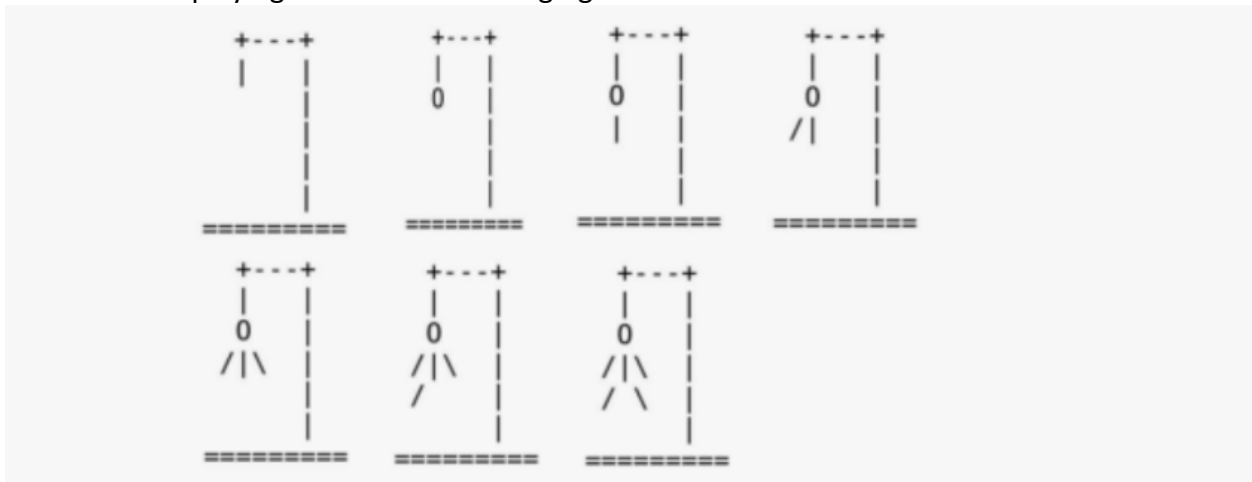
### Question3

For this assignment, you will be making a simple version of the word game [Hangman](#). This game will be played in the terminal window. In this game, the computer picks a random secret word from an available list. The player then guesses a letter that they think may appear in the word. The computer checks whether the letter appears in the word and either reveals *all* instances of it (if it exists) or takes away one of the player's lives. The computer then displays the progress that the player has made towards guessing the word. The game continues until either all of the letters have been guessed (and the player wins) or all of the player's lives are used up (and the player loses). In either case, the computer should announce this event and the game should end.

You will add a list of 100 words in your program.

Example list of words "fast, programming, students, are, genius"

You will be displaying one of the following figures after round.



Following additional rules will be used to play the game.

1. You will ask the player to input his/her name
2. If there is a common alphabet in the user's name and the secret word
  - a. And the player gets it right, he/she will get an additional life. If he/she has not previously lost any life he/she will not get any advantage
  - b. And the player gets it wrong then he/she will lose 2 lives instead of one.

You must have a create function to create a list of words, draw function to show hangman, hangwoman, decision function to show the result.

#### Question 4

You are required to implement “Find/Replace” tool found in a typical text editor. Your program will ask user to enter a text of max length 500 characters. Then the program will ask which word the user want to find in the text. Followed by which word should be used to replace the word. In the end the program will display the resultant text. The program will continue to take request from the user unless user enters empty word to find in the text.

Consider the following example

Please enter the text (500 chars max)

You are required to implement “Find/Replace” tool found in a typical text editor. Your program will ask user to enter a text of max length 500 characters. Then the program will ask which word the user want to find in the text. Followed by which word should be used to replace the word. In the end the program will display the resultant text. The program will continue to take request from the user unless user enters empty word to find in the text.

Please enter the word to find

text

Please enter the replace word

figure

Result

You are required to implement “Find/Replace” tool found in a typical figure editor. Your program will ask user to enter a figure of max length 500 characters. Then the program will ask which word the user want to find in the figure. Followed by which word should be used to replace the word. In the end the program will display the resultant figure. The program will continue to take request from the user unless user enters empty word to find in the figure

#### Question 5:

Your friend came to you for spending his/her holidays and has some idea in a mind of game and he/she knows you are doing CS degree so interested in development of that game. The game name is "Gain and Win" game. Your task is to create that game for your friend which should include three functions: **draw**, **play**, and **showOutput**. The objective is to move through a grid and accumulate points, but if you land on a cell with a value of 0, your score resets to zero and the game ends when score reaches to 100.

Here are the specific requirements for each function:

##### 1. Function draw

- **Arguments:** **n** (number of rows) and **m** (number of columns) where  $1 < n < 100$  and  $1 < m < 150$ .
- **Output:** Generate  $n \times m$  game board filled with random values between 0 and m. You can use standard C++ randomization functions to generate these values. The board should be displayed on the console.

##### 2. Function play

- **Argument:** **s**, where s is between 0 and n.
- **Execution:** Start the game at cell (s, 0). The player picks the value from this cell, denoted as 'q'.
  - Move to cell (k, q), where 'k' starts at 0 and increments with each jump.
  - If 'q' reaches n, reset it to 0.
  - If the cell value at (k, q) is 0, reset the player's score to zero.
  - Otherwise, add the cell value to the player's score.
- **Exit Condition:** The game ends when the player's score crosses 100.

##### 3. Function showOutput

- **Output:** At the end of the game display the final score and the number of jumps made during the game.

Your program should allow the user to input the values for 'n' and 'm' at the start of the program. It should also prompt the user for the initial value 's'.

#### Note:

- You may use standard C++ libraries for generating random numbers.

- The program should handle inputs within the specified ranges gracefully and report any invalid inputs.

Your task is to write the C++ program that implements these three functions and fulfills the requirements. Ensure that your code is well-documented and organized.

**Important:** Remember to follow C++ coding conventions and best practices in your solution.



### Question 6

Write a C++ program to take 3 arrays as an input from the user and pass these arrays to the function name "order". The function "order" will store the values of these 3 arrays in such an order that even values appear first and then all the odd values.

The arrays may contain repetitive values and you have to remove the repetitive values and in order to fill that space in the resultant array.

- -1 will be inserted in the start of the resultant array if the repetitive value is occurred in array1,
- -2 will be inserted in the start of the resultant array if the repetitive value is occurred in array2
- -3 will be inserted in the start of the resultant array if the repetitive value is occurred in array3
- If the repetition occurs within array1 and array2 so -12 value will be inserted in the resultant array.
- If the repetition occurs within array2 and array3, -23 value will be inserted in the resultant array
- If the repetition occurs within array1 and array3 so -13 value will be inserted in the resultant array.
- And If the repetition occurs within array1, array2 and array3 so -123 value will be inserted in the resultant array.
- The size of the resultant array should be the sum of all the input arrays size

In the resultant array these –ve values should be appear in ascending order at the start of resultant array.

Note: In all the three arrays if user enter any –ve value you have to skip that value and prompt for a +ve value.