National University of Computer and Emerging Sciences

FAST School of Computing **Fall-2023** **Islamabad Campus**

CS-1002: Programming Fundamentals Lab

Serial No:

Final Exam

Total Time: 3 Hours

Total Marks: 100

Wednesday, 3rd January, 2024

Course Instructor

Ammar Masood

_____

Signature of Invigilator

Student Name	**Roll No.**	**Course Section**	**Student Signature**

DO NOT OPEN THE QUESTION BOOK OR START UNTIL INSTRUCTED.

Instructions:

1. Attempt the questions on the provided system. Attempt all of them. Read the question carefully, understand the question, and then attempt it. Write two lines of PF Lab feedback on page 2 to get a bonus of two marks. Write clean code, correct indentation & make use of good programming practices. "THINK MORE CODE LESS"
2. After being asked to commence the exam, please verify that you have **nine (9)** different printed pages including this title page. There are a total of **4** questions.
3. **Please make sure to submit a zipped folder(.zip) named *yourRollNumber (23xxxx).* Please note that there is NO dash(-) and alphabet(i) in the folder name. The folder should contain only .cpp files. The questions must be named as q1.cpp, q2.cpp…**
4. Submission will only be accepted on Google Classroom. Failure to submit according to submission guidelines will result in deduction of 20% marks in your section Tab.
5. Late submission will not be accepted.
6. Please note that you have to write the submission time at GCR on the front page of the question paper.

	Q-1	**Q-2**	**Q-3**	**Q-4**	**Total**
Marks Obtained					
Total Marks	**25**	**25**	**25**	**25**	**100**

Question 1 [25 Marks]

Question: Write a C++ **function** to draw the following pattern using loop(s). **YOU MUST FOLLOW THE FOLLOWING STEPS**

1. Make a function named "**pattern**" that takes two integer arguments i.e **waveLength & waveHeight.** The function will print the pattern.

```
void function1(int waveLength,int waveHeight)
```

2. Your **main** function should look like the following.

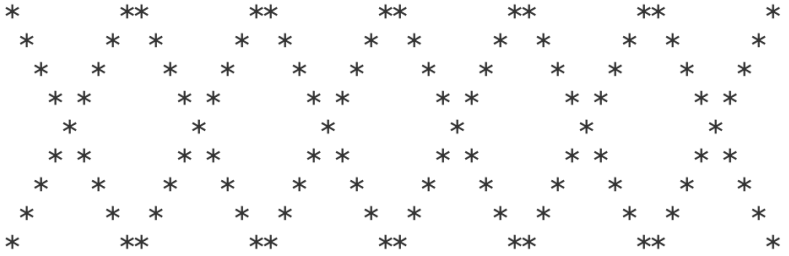
```
int main(){
    for (int i = 3, j= 3; i<=8; i++,j+=2){
        cout<< "length="<< i << " & height=" << j << endl;
        pattern(i,j);
        cout<<endl;
    }
}
```

3. The output should look like this, if it's **100%** correct mark yourself **25/25** otherwise **0/25. Even an error of a single character will be considered incorrect. If you** mark yourself fairly you will get **1.5 marks.**

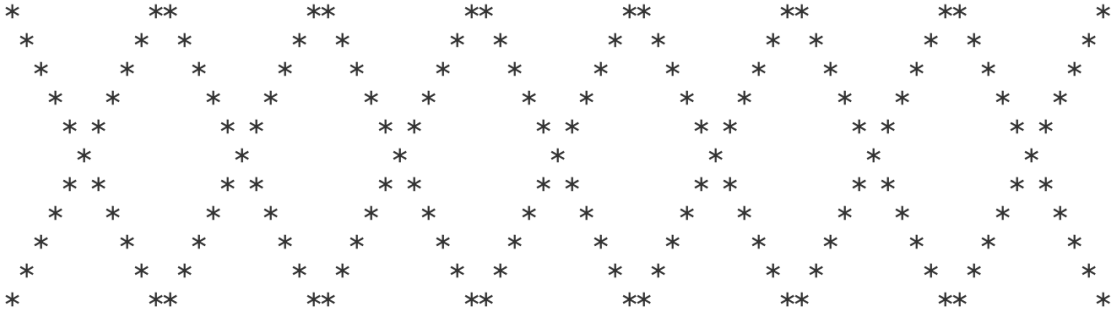
Student's Evaluation (Encircle your option)			**For Examiner Use Only**			
25	**zero**		**26.5**	**25**	**1.5**	**0**

```
length=3 & height=3
* ** ** *
 *  *  *
* ** ** *

length=4 & height=5
*   **   **   **   *
 * *  * *  * *  * *
  *    *    *    *
 * *  * *  * *  * *
*   **   **   **   *

length=5 & height=7
*     **     **     **     **     *
 *   *  *   *  *   *  *   *  *   *
  * *    * *    * *    * *    * *
   *      *      *      *      *
  * *    * *    * *    * *    * *
 *   *  *   *  *   *  *   *  *   *
*     **     **     **     **     *
```

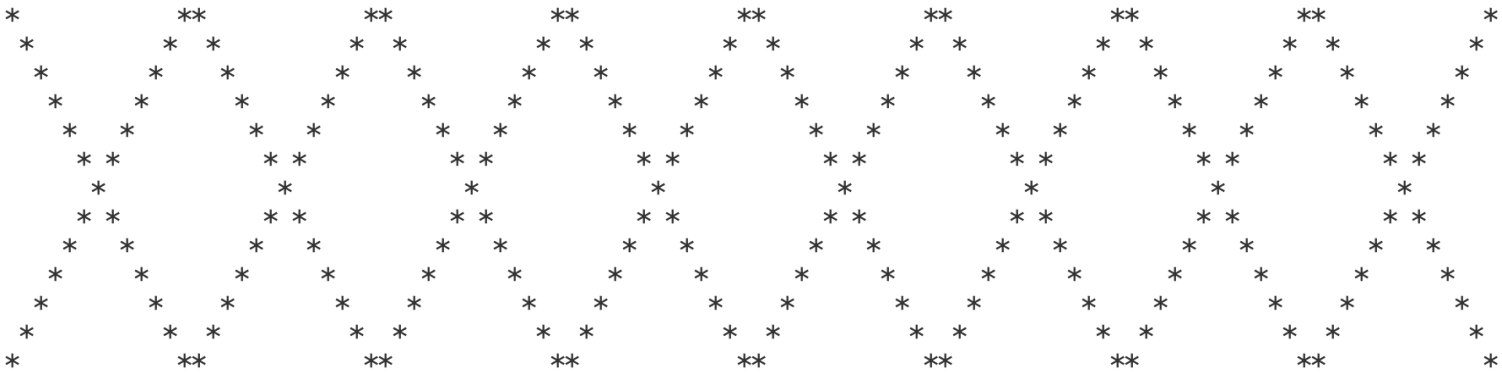
length=6 & height=9



length=7 & height=11



length=8 & height=13



Question 2 [25 Marks]

Question**: Write a C++ **function** signature as **void magicSquare(const integer sqaureDimension)** for generating magic **squares** of any **odd size. Call this function 4 times in the main function by passing arguments 3,5,7 & 9 respectively.

An 3x3 normal magic square is an arrangement of the numbers 1, 2, 3, ... n2 in a square array, with the property that the sum of every row and column, as well as both diagonals, is the same number. An example of a 3x3 normal magic square is

6 1 8
7 5 3
2 9 4

You can verify that each of the three rows, the three columns, and the two diagonals add to 15.
***Algorithm**: To build a magic square we will be using Loubere's algorithm which is as follows:*

Step 1: Begin by placing a 1 in the middle location of the top row:

	1	

Step 2: We then write successive integers in an upward-right diagonal path (i.e. in the right column of row above current inserted element position), with the following special cases:

a) When this upward-right movement would result in a location outside the boundaries of the square, we place the new number at the opposite end of the row or column that would contain the new number, if the rows and columns were not bounded.
b) If the upward-right square is already occupied, place the new number directly below the current one.

In the 3x3 case, this gives us the following sequence of placements:
1.Place 1 in the middle location of the top row.

	1	

2. Next place 2 on the above row and right column (Step 2 a). Since there is no row above first row so we place the 2 in the last row and next column.

	1	
		2

3. Next place 3 on the above row and right column (Step 2 a). Since there is no right column row so we place the 3 in the previous row and first column.

	1	
3		
		2

4. Next place 4 on the above row and right column (Step 2 a). Since this place is already occupied place 4 below 3 (Step 2 b).

	1	
3		
4		2

5. Next place 5 on the above row and right column of number 4 (Step 2 a).

	1	
3	5	
4		2

6. Next place 6 on the above row and right column of number 5 (Step 2 a).

	1	6
3	5	7
4		2

7. Next place 7 on the above row and right column of number 6 (Step 2 a). Since this place is already occupied by 4 place 7 below 6.

	1	6
3	5	7
4		2

8. Next place 8 on the above row and right column of number 7 (Step 2 a). Since this place is already outside the boundaries, place 8 in the first row and first column.

8	1	6
3	5	7
4		2

9. Finally, place 9 on the above row and right column of number 8 (Step 2 a). Since this place is already outside the boundaries as there is no row above first row so we place the 9 in the last row and next column to complete our magic square.

8	1	6
3	5	7
4	9	2

You need to RUN your function for arguments = 3,5,7,9

dimension = 3	dimension = 5
8 1 6 3 5 7 4 9 2	17 24 1 8 15 23 5 7 14 16 4 6 13 20 22 10 12 19 21 3 11 18 25 2 9
dimension = 7	**dimension = 9**
30 39 48 1 10 19 28 38 47 7 9 18 27 29 46 6 8 17 26 35 37 5 14 16 25 34 36 45 13 15 24 33 42 44 4 21 23 32 41 43 3 12 22 31 40 49 2 11 20	47 58 69 80 1 12 23 34 45 57 68 79 9 11 22 33 44 46 67 78 8 10 21 32 43 54 56 77 7 18 20 31 42 53 55 66 6 17 19 30 41 52 63 65 76 16 27 29 40 51 62 64 75 5 26 28 39 50 61 72 74 4 15 36 38 49 60 71 73 3 14 25 37 48 59 70 81 2 13 24 35

The output should look like this, if it's **100%** correct mark yourself **25/25** otherwise **0/25. Even an error of a single digit will be considered incorrect.** If you mark yourself fairly you will get a 1.5 marks **bonus.**

Student's Evaluation (Encircle your option)			**For Examiner Use Only**			
25	**zero**		**26.5**	**25**	**1.5**	**0**

Question 3 [25 Marks]

Question: You are tasked with developing a simple user registration and login system in C++.

- Implement a displayMenu() function that has the following menu options. This displayMenu() will be called repeat in the main function until the user exits the program.

    - ❖ Press 1 to Register => registerUser function should be called by pressing 1
    - ❖ Press 2 to Login => LoginUser function should be called by pressing 2
    - ❖ Press 3 to Exit => It will terminate the displayMenu() function call

- Implement a registerUser() function that will ask the user to enter his name, email, username, password, and age. After taking an input, call the **isPasswordStrong(string)** to validate if the password is strong or not. If the password is strong, the registration details should be stored in a text file with the given format. For example if I enter my name then **"ammar.txt"** file will be created with all the entered information. Similarly, if I register another user a new file for example **"ahmed.txt"** will be created containing user info according to the following format.

ammar.txt
name:ammar **age:24** **email:ammar.masood@isb.nu.edu.pk** **username:ammar_user** **password:12am34**

- Implement a password validation C++ function **isPasswordStrong(string)** for a user registration system. The password must adhere to the following criteria.
    - ❖ At least 8 characters long.
    - ❖ Contains a mix of uppercase and lowercase letters.
    - ❖ Includes at least one numeric digit.
    - ❖ Contains at least one special character (e.g., @, #, $, %).
- LoginUser() can let users log in by entering their name, username and password. If the entered credentials match the data stored in the text file, display all the user details.There is no need for input validation.

For Examiner Use only		
Code Indentation & Readability (complete)	3	
Correctness (complete program)	12	
Implementation (1+3+3+3)	10	

Question 4 [25 Marks]

Question: Write a C++ **function** named replaceSubstring(string,string,string) that takes three strings (sentence, find, replace), replaces a Find substring with a Replace substring and creates & displays a new string. For example:

There is no need to take an input, just call the functions with hardcoded arguments.

String: "I am Pakistani so I support the Pakistani Cricket team in Pak-India matches."
Find Substring: Pak
Replace Substring: Afghan
New String: "I am Afghanistani so I support Afghanistani Cricket team in Afghan-India matches."

String: "I am Pakistani so I support Pakistani Cricket team in Pak-India matches."
Find Substring: Pakii
Replace Substring: Afghan
New String: "I am Pakistani so I support Pakistani Cricket team in Pak-India matches." (as Pakii does not exist in the string).

For Examiner Use only		
Code Indentation & Readability (complete program)	3	
Correctness (complete program)	14	
Implementation	8	