SZABIST UNIVERSITY

SHAHEED ZULFIKAR ALI BHUTTO INSTITUTE OF SCIENCE AND TECHNOLOGY UNIVERSITY

CS

Group No: 4 -5 **Hackathon'24** **Marks: 20**

Section:

FUNDAMENTALS OF PROGRAMMING

Task 1: (2 Marks)

Create a program using nested loops to print following pattern:

```
1******
12*****
123****
1234***
12345**
123456*
1234567
```

Task 2: (3 Marks)

Write a program that simulates a lottery. The program should have an array of 5 integers named winningDigits, with a randomly generated number in the range of 0 through 9 for each element in the array. The program should ask the user to enter 5 digits and should store them in a second integer array named player. The program must compare the corresponding elements in the two arrays and count how many digits match. For example, the following shows the winningDigits array and the Player array with sample numbers stored in each. There are two matching digits, elements 2 and 4.

WinningDigits	7	4	9	1	3
Player	4	2	9	7	3

Once the user has entered a set of numbers, the program should display the winning digits and the player's digits and tell how many digits matched.

Task 3: (4 Marks)

Develop a C program for a Music Playlist System that utilizes structures to manage songs in a playlist. The program should allow users to add songs, play songs, and display the playlist. Implement file handling to persistently store and retrieve song information. Song records are saved in a file named "playlist_records.txt." The structure for a song is defined as follows:

```
struct Song {
int songID;
char title[100];
char artist[100];

float duration; // in minutes

};
```

The program provides a menu-driven interface with the following options:

Add Song

Play Song

Display Playlist

Exit

The program should be able to handle a reasonable number of songs. Ensure that it loads song records from the "playlist_records.txt" file at the start and saves records to the same file before exiting. Implement proper error handling for file operations.

Provide the complete C code for the program, including the necessary functions for adding songs, playing songs, displaying the playlist, and handling file operations. Test your program by adding songs, playing songs, and viewing the playlist to ensure that the file handling functions work correctly.

Note: Pay attention to details such as reading and writing data in the correct format, handling situations where the file may not exist or is not accessible, and ensuring that the playlist is displayed accurately.