National University of Computer and Emerging Sciences

Assignment No. 3

Programming Fundamentals

CS1002

Fall 2022

Deadline: 31-10-2022, 11:59PM

Department of Computer Science

Submission Instructions:

- All problems must be solved by following the order and submitted in .cpp file in zip folder and also have to submit the word file (.docx) otherwise marks will be deducted.
- Printed assignment will get you zero marks.
- This is an individual assignment.
- **Plagiarism is strictly prohibited.**

Assignment Tasks

1. Write a C++ code that takes an integer **n** as an input from the user which is **greater** than **5** else print **Incorrect Input!** If **n** multiple of **5** then take a variable temp = n/2 and print temp times **n** else take a variable temp = n/3 and print temp times **n**.

   Example:

   Input: 10
   Output: 1010101010

   Input: 13
   Output: 13131313

   Input: 4
   Output: Incorrect Input!

2. Write a C++ program that takes an integer type variables from the user and output the following things;
   1. **Count** the number of digits in that number
   2. **Reverse** this number and store back into the original variable.

   Example:

   Input: 37856
   Output: **5** and 65873

   Input: 1050809
   Output: **7** and 9080501

3. Write a C++ code that inputs an **integer** number of any length from user and print it into **words**.

   Example:

   Input: 8745
   Output: Eight Seven Four Five

4. Write a C++ program to prompt user to enter **month** and **year**. Your program should display the number of days in the month of that year.

<u>Example:</u>

Input: 2 and 2000
Output: 29 days

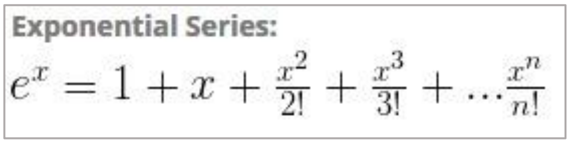
5. Write a C++ code that takes a binary input from the user and contains more than **six** and less than **ten** digits and counts the total **1's** present in it. If the digits are less than **six** it will print **Digits are less than six** and if the digits are greater than 10 it will print **Digits are greater than 10**. In case if the user has entered a number that is not a binary number than it will print **Incorrect Input! It is not a Binary Number**.

<u>Example:</u>

Input: 10011001
Output: Digits are greater than six!
Total no. of 1's are : 4 ones

6. In this question, find the **Sum** of following exponential series up to **N** terms. Ask the user to input **N** and **X** from the console. You are not allowed to use any **built-in, library or custom**-made functions.

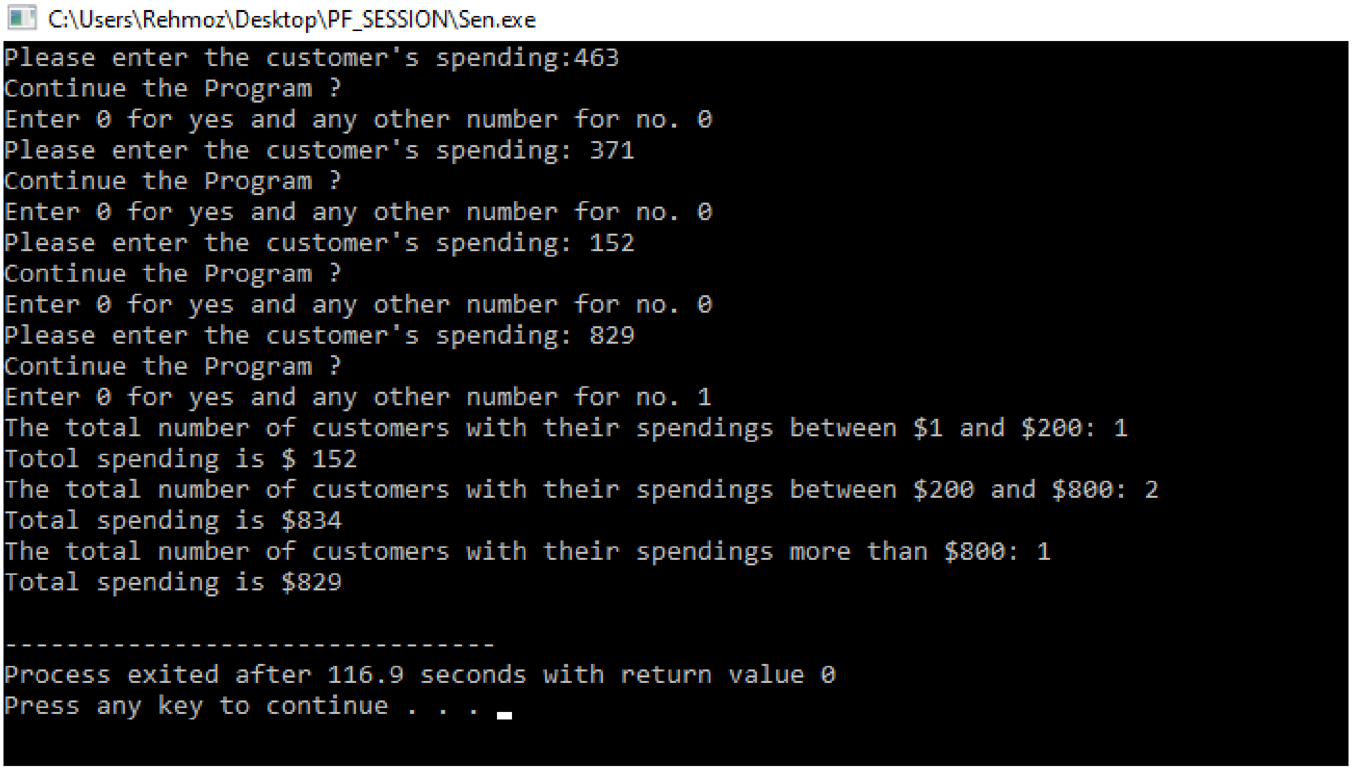
Hint: you can calculate the **power** and **factorial** in separated loops and use them to find the sum!



Exponential Series:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots \frac{x^n}{n!}$$

7. Write a C++ Program that is using a sentinel control while loop statement and if statement to write a program that can be used at a department store's register the records the number of customers who spent between $0.00-$200.00, and the number of customers who spent between $200.01-$800.01 and the number of customers who spent $800.01 or more. At the end of the day, the program will print out the number of customers in each of categories and total spending in each categories. The output will be like this:



```
Please enter the customer's spending:463
Continue the Program ?
Enter 0 for yes and any other number for no. 0
Please enter the customer's spending: 371
Continue the Program ?
Enter 0 for yes and any other number for no. 0
Please enter the customer's spending: 152
Continue the Program ?
Enter 0 for yes and any other number for no. 0
Please enter the customer's spending: 829
Continue the Program ?
Enter 0 for yes and any other number for no. 1
The total number of customers with their spendings between $1 and $200: 1
Totol spending is $ 152
The total number of customers with their spendings between $200 and $800: 2
Total spending is $834
The total number of customers with their spendings more than $800: 1
Total spending is $829

--------------------------------
Process exited after 116.9 seconds with return value 0
Press any key to continue . . .
```

8. An organization wanted to give **bonus** to his employees in **three** situations

a) For either gender, if age is more than 30 and basic pay is more than 25000 then bonus is 25% of basic pay

b) For male, if age is less than 30, and basic pay is more than 21000 then bonus is 17% of basic pay

c) For female employees, if age is less than 25, and basic pay is more than 18000 then bonus is 13% of basic pay

In any other condition organization will not give bonus.

Use sentinel controlled while loop to input the employees data from user. Whenever user enters, **-999** then stop your program!

Program-Input: basic pay, gender, age, and calculate the bonus for employees.

9. Snakes and ladders is an ancient Indian board game that's regarded today as a worldwide classic. It requires two or more players and takes place on a board with numbered, gridded squares. Throughout the board, there are snakes and ladders, which connect different squares. Players **roll a dice** and navigate the board. Landing on a ladder advances a player to a square further up the board, while landing on a snake means they have to go back to a previous square.

Note: The aim of the game is to reach the final square (**100**). The game is a race that's based on sheer luck, and is popular with children.

Create a **main menu**, which will display the menu of the game the menu of the game will consist of;

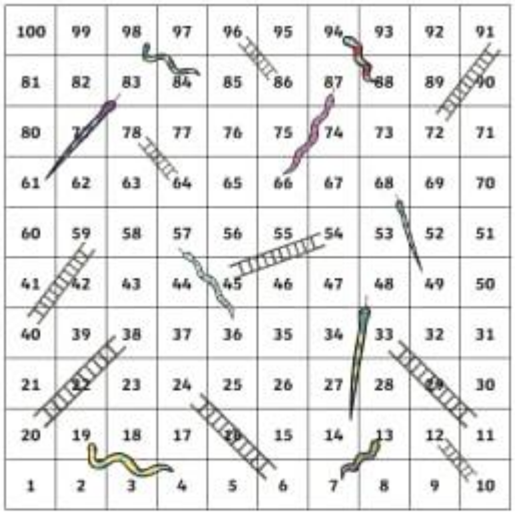
- **Instructions/rules** to be displayed
- **Play_game** (This menu will further gave me choice to select between the choice of user-1 or user-2 by rolling a dice. (Use random function to roll a dice.)
- **BOARD** will start from **1-100**
- The reach destination must be set to **100**
- You have to place **5 snakes** and **7 Ladders** in the game.
- The **dice** can minimum roll up to **one** and maximum to **six**.

Hint: if (n == 17 or n==32 or n==55 or n==72 or n==98) //conditions for snake then just decrement **'n' some fix value** depending upon the condition**.**

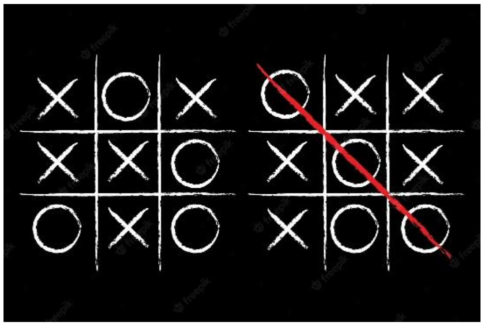
Reminder: No Board is required while making the game. Also, you are not allowed to use any built in functions, any function structures, arrays and also the libraries.

How to Play Snakes and Ladders

- Each player begins with a token or figure on the '1' square in the bottom left-hand corner of the board.
- Players take turn rolling a die and moving their token in accordance with the number, they roll. If they roll a one they move one square; if they roll a two they move two squares, and so on.
- If a player lands on the lower end of a ladder, they 'climb' that ladder to the square that features its top end.
- Conversely, if players land on the higher-numbered square of a snake, they fall down to its lowest-numbered square. If a player rolls a six, they get another roll after moving their token. To speed up the game, players can use two dice.
- Whoever reaches the last square first is the winner.



10. Write a C++ program to create a **Tic Tac Toe** game. Tic-tac-toe is a game in which two players take turns in drawing either an **'O'** or an **'X'** in a square of a grid consisting of **nine** squares. The winner is the first player to get three of the same symbols in a row, column or any of the diagonal.



- ➢ Your Program should execute for one modules of the game.
  - **Multiplayer (Player Vs Player)**

Your code should base on the following concept of C++ language.

- ➢ Create a **main menu** which will display the menu of game ,the menu of game will consist of ;
  - **Instructions/rules** to be displayed
  - **Play_game** (This menu will further gave me choice to select between the choice of user-1 or user-2.
  - **THE BOARD SIZE WILL BE 3x3.**

Reminder: Board is required while making the game. Also, you are not allowed to use any built in functions, any function structures, arrays and also the libraries.

GAME PLAY

The goal of tic-tac-toe is to be the first player to get three symbols in a row on a 3-by-3 grid. To start, one player draws a board, creating a grid of squares, **3-by-3** in this task.

In a 3-by-3 grid game, players alternate placing **Xs** and **Os** on the board until either player has three symbol in a row, horizontally, vertically, or diagonally or until all squares on the grid are filled. If a player is able to draw three Xs or three Os in a row, then that player wins. If all squares are filled and neither player has made a complete row of Xs or Os, then the game is a draw.

- Your game will start from a toss, you have to create a Bool variable as **bool toss** (who won the toss to have the first turn.)
- Game will not end until someone won or game is drawn.
- You will create a **winning condition.** In this, you will check all the possible combination of winning condition. Similarly you will create **lose_condition & draw_condition** .
- You have to create a **bool player_turn, which** will change the player turn after every valid move.

Note: Proper code indentation will hold extra marks!