INSTITUTE OF TECHNOLOGY AND MANAGEMENT
SKILLS UNIVERSITY,
KHARGHAR, NAVI MUMBAI

C++ PROGRAMMING LAB

Prepared by:

Name of Student: _Shikha singh_____

Roll No: __25_____

Batch: 2023-27

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Ex	List of Experiment

p. No	
1	Write a program to find the roots of a quadratic equation.
2	Write a program to calculate the power of a number using a loop.
3	Write a program to check if a given string, is a palindrome.
4	Write a program that simulates a simple ATM machine, allowing users to check their balance, deposit, or withdraw money using a switch statement.
5	Write a program that finds the largest among three numbers using nested if-else statements
6	Write a program that determines the grade of a student based on their marks of 5 subjects using if-else-if ladder.
7	Write a program to find the sum of digits of a number until it becomes a single digit number.
8	Write a program to print a Pascal's triangle using nested loops.
9	Write a program to calculate the sum of series 1/1! + 2/2! + 3/3! + ... + N/N! using nested loops.
10	Write a program to create an array of strings and display them in alphabetical order.
11	Write a program that checks if an array is sorted in ascending order.
12	Write a program to calculate the sum of elements in each row of a matrix.
13	Write a program to generate all possible permutations of a string.
14	Create a C++ program to print the following pattern: ***** * * * * * *

	*****

15	Write a C++ program to display the following pattern: 1 232 34543 4567654 34543 232

16	Write a program to creating an inventory management system for a small store. The system should use object-oriented principles in C++. Yourprogram should have the following features:
	• Create a **Product** class that represents a product in the inventory. Each **Product** object should have the following attributes:
	  • Product ID (an integer)
	  • Product Name (a string)
	  • Price (a floating-point number)
	  • Quantity in stock (an integer)
	• Implement a parameterized constructor for the **Product** class to initialize the attributes when a new product is added to the inventory.

17	Write a program to manage student records. Create a class Student with attributes such as name, roll number, and marks. Implement methods for displaying student details, adding new students, and calculating the average marks of all students in the record system.
18	Write a program that implements a basic calculator. Use a class Calculator with methods to perform addition, subtraction, multiplication, and division of two numbers. The program should allow the user to input two numbers and select an operation to perform.
19	Write a program to simulate a simple online shop. Create a class Product with attributes like name, price, and quantity in stock. Implement methods for adding products to the shopping cart, calculating the total cost, and displaying the contents of the cart.

20	Write a program to manage student grades for a classroom. Create a class Student with attributes for student name and an array to store grades. Implement methods for adding grades, calculating the average grade, and displaying the student's name and grades. Use constructors and destructors to initialize and release resources.

Name of Student: Shikha singh

_____ Roll Number: 25

_____ Experiment No:

11

Title: 11. Write a program that checks if an array is sorted in ascending order.

Theory:

Input:

- **The user is prompted to enter elements into the array.**
- **Sorting Check:**
- **The program checks whether the entered array is sorted in ascending order.**
- **Output:**
- **The program prints a message indicating whether the array is sorted or not.**
- **Result:**
- **If the array is sorted, the program prints "The array is sorted in ascending order."**
- **If the array is not sorted, the program prints "The array is not sorted in ascending order."**
- **Limitation:**
- **This program assumes that the user enters valid integer inputs. It doesn't handle non-integer inputs or other edge cases.**
- **Extension:**
- **This basic program can be extended to handle various input types, provide more detailed information about the sorting status, and include additional sorting algorithms for comparison.**

Code:

```
#include <iostream>
using namespace std;
int main() {
   const int size = 5;
   int arr[size];
   cout << "Enter " << size << " elements into the array:\n";
   for (int i = 0; i < size; ++i) {
      cout << "Enter element " << i + 1 << ": ";
      cin >> arr[i];
   }
   bool isSorted = true;
   for (int i = 1; i < size; ++i) {
      if (arr[i] < arr[i - 1]) {
         isSorted = false;
         break;
      }
   }
   if (isSorted) {
      cout << "The array is sorted in ascending order.\n";
   } else {
      cout << "The array is not sorted in ascending order.\n";
   }
   return 0;
}
```

Output: (screenshot)

Test Case: Any two (screenshot)

```
  cd "/Users/shikhasingh/Desktop/C++"
  cd "/Users/shikhasingh/Desktop/C++/" && g++ 11.ascendingorder.cpp -o 11.ascendingorder && "/Users/shikhasingh/
  Desktop/C++/"11.ascendingorder
● shikhasingh@SHIKHAs-MacBook-Air C++ % cd "/Users/shikhasingh/Desktop/C++"
● shikhasingh@SHIKHAs-MacBook-Air C++ % cd "/Users/shikhasingh/Desktop/C++/" && g++ 11.ascendingorder.cpp -o 11.
  ascendingorder && "/Users/shikhasingh/Desktop/C++/"11.ascendingorder
  Enter 5 elements into the array:
  Enter element 1: 67
  Enter element 2: 45
  Enter element 3: 5
  Enter element 4: 89
  Enter element 5: 76
  The array is not sorted in ascending order.
○ shikhasingh@SHIKHAs-MacBook-Air C++ %
```

```
  cd "/Users/shikhasingh/Desktop/C++"
  cd "/Users/shikhasingh/Desktop/C++/" && g++ 11.ascendingorder.cpp -o 11.ascendingorder && "/Users/shikhasingh/
  Desktop/C++/"11.ascendingorder
● shikhasingh@SHIKHAs-MacBook-Air C++ % cd "/Users/shikhasingh/Desktop/C++"
● shikhasingh@SHIKHAs-MacBook-Air C++ % cd "/Users/shikhasingh/Desktop/C++/" && g++ 11.ascendingorder.cpp -o 11.
  ascendingorder && "/Users/shikhasingh/Desktop/C++/"11.ascendingorder
  Enter 5 elements into the array:
  Enter element 1: 56
  Enter element 2: 79
  Enter element 3: 88
  Enter element 4: 91
  Enter element 5: 92
  The array is sorted in ascending order.
○ shikhasingh@SHIKHAs-MacBook-Air C++ %
```

Conclusion:

In conclusion, the program is a basic

implementation to check if an array is sorted

in ascending order. Understanding such

fundamental concepts is crucial for building

a strong foundation in programming.