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

xp. No	List of Experiment
	Write a program to find the roots of a quadratic equation.
	Write a program to calculate the power of a number using a loop.
	Write a program to check if a given string, is a palindrome.
	Write a program that simulates a simple ATM machine, allowing users to check their balance, deposit, or withdraw money using a switch statement.
	Write a program that finds the largest among three numbers using nested if-else statements
	Write a program that determines the grade of a student based on their marks of 5 subjects using if-else-if ladder.
	Write a program to find the sum of digits of a number until it becomes a single-digit number.
	Write a program to print a Pascal's triangle using nested loops.
	Write a program to calculate the sum of series 1/1! + 2/2! + 3/3! + ... + N/N! using nested loops.
0	Write a program to create an array of strings and display them in alphabetical order.
1	Write a program that checks if an array is sorted in ascending order.
2	Write a program to calculate the sum of elements in each row of a matrix.
3	Write a program to generate all possible permutations of a string.
4	Create a C++ program to print the following pattern: ***** * * * * * * *****
5	Write a C++ program to display the following pattern: 1 232

	34543 4567654 34543 232
6	Write a program to creating an inventory management system for a small store. The system should use object-oriented principles in C++. Your program 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.
7	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.
8	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.
9	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.
0	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: 13

Title:13. Write a program to generate all possible permutations of a string.

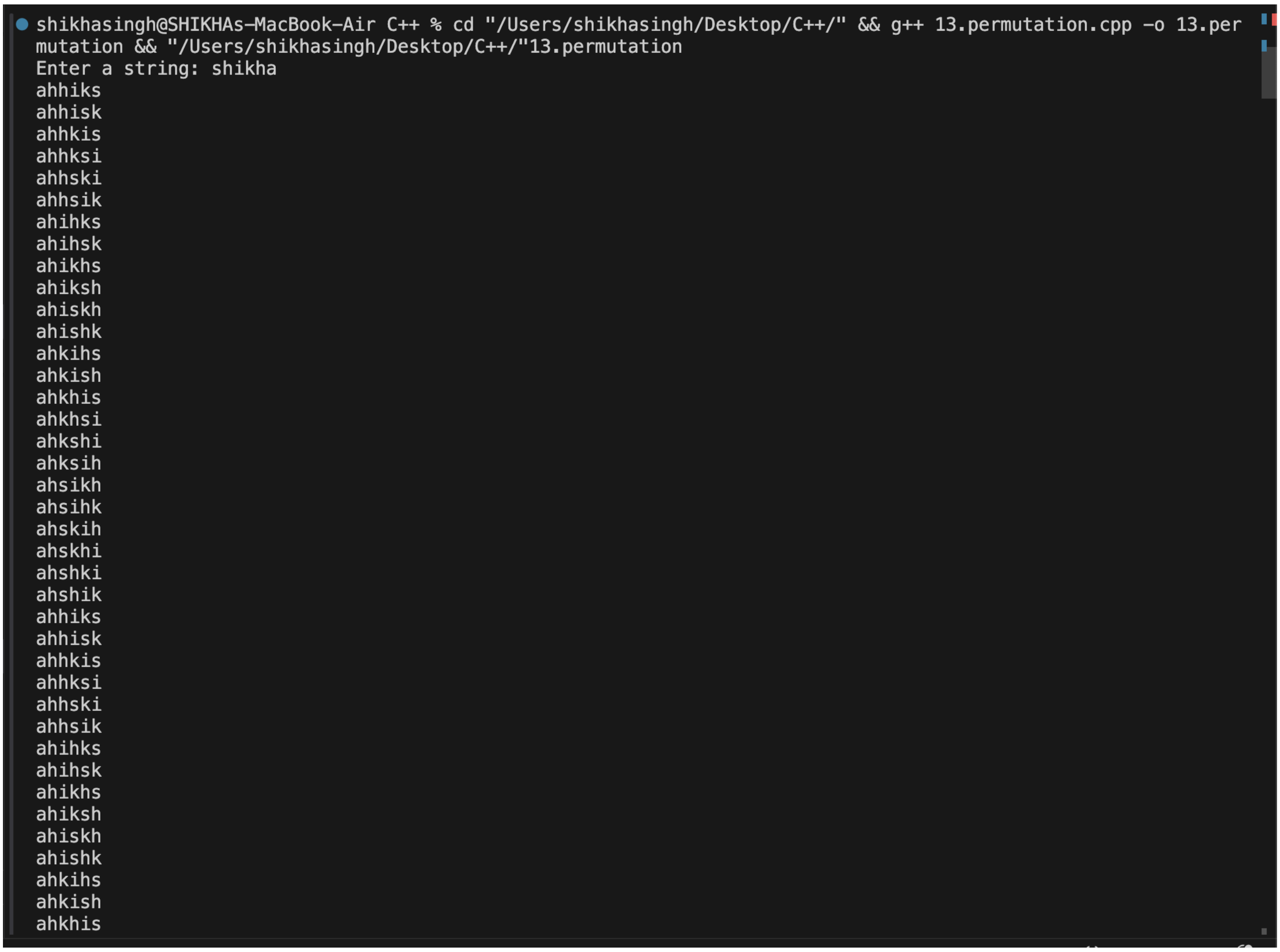
Theory:In mathematics, a permutation of a set is an arrangement of its members into a sequence or linear order. In the context of strings, a permutation is all possible arrangements of its characters. For a string of length n and, there are n! permutations.

The program to generate all permutations typically uses recursion and swaps elements in the string to achieve different arrangements. The recursive function would fix one character at a time and generate permutations for the remaining characters.

Code:

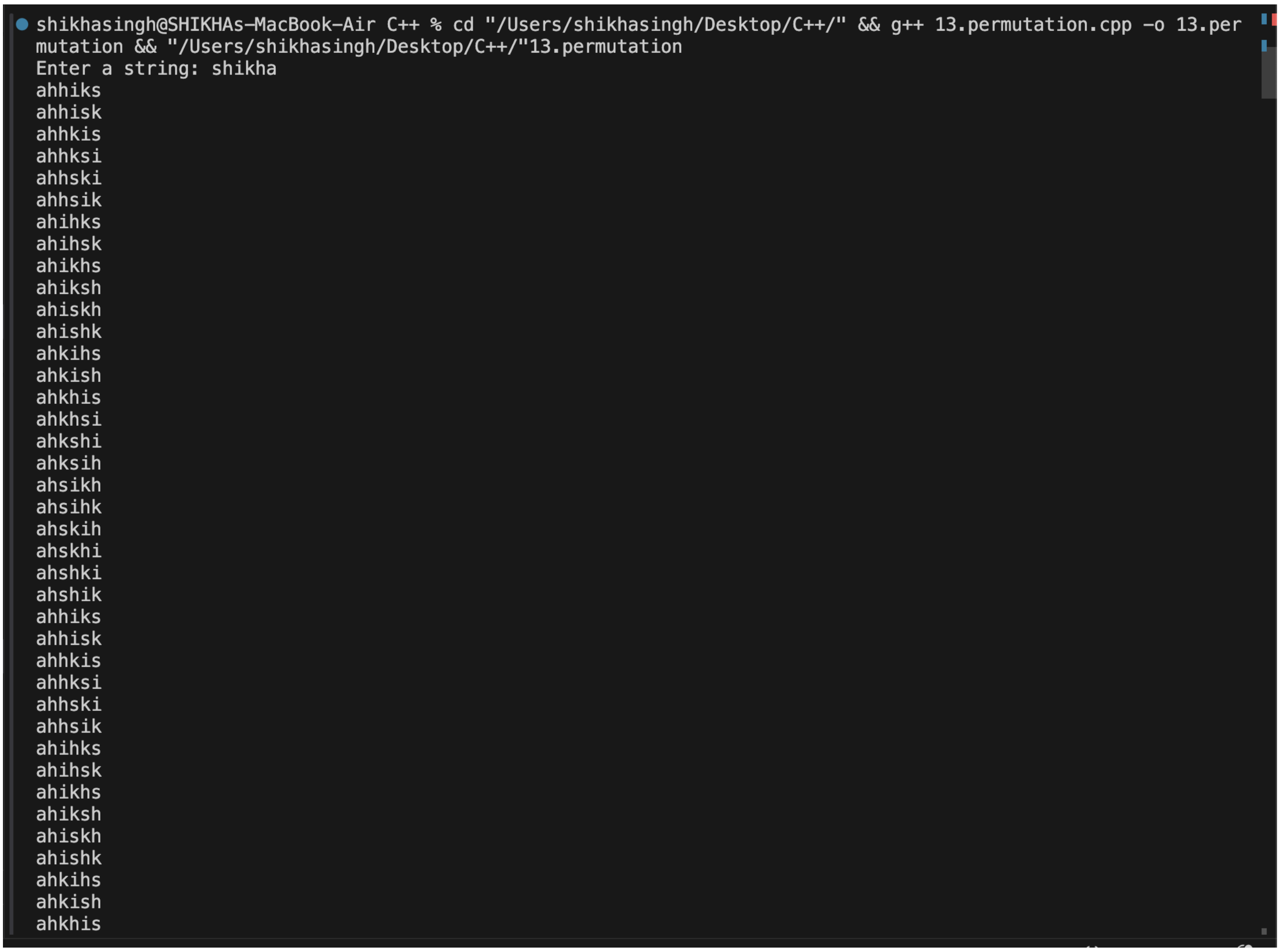
```
#include <iostream>
#include <algorithm>
using namespace std;
void generatepermutations(string& str, int start, int end)
{
   if (start == end) {
      cout << str << endl;
       return;
   }
   for (int i = start; i <= end; ++i) {
       swap(str[start], str[i]);
       generatepermutations(str, start + 1, end);
       swap(str[start], str[i]);
   }
}
int main() {
  string input;
   cout << "Enter a string: ";
   getline(cin, input);
   sort(input.begin(), input.end());
   generatepermutations(input, 0, input.length() - 1);
   return 0;
}
```

Output: (screenshot)

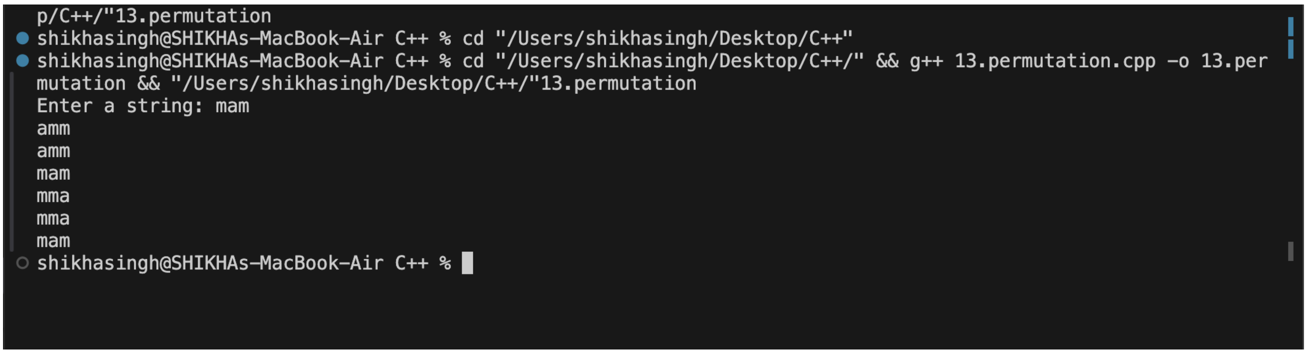


```
shikhasingh@SHIKHAs-MacBook-Air C++ % cd "/Users/shikhasingh/Desktop/C++/" && g++ 13.permutation.cpp -o 13.per
mutation && "/Users/shikhasingh/Desktop/C++/"13.permutation
Enter a string: shikha
ahhiks
ahhisk
ahhkis
ahhksi
ahhski
ahhsik
ahihks
ahihsk
ahikhs
ahiksh
ahiskh
ahishk
ahkihs
ahkish
ahkhis
ahkhsi
ahkshi
ahksih
ahsikh
ahsihk
ahskih
ahskhi
ahshki
ahshik
ahhiks
ahhisk
ahhkis
ahhksi
ahhski
ahhsik
ahihks
ahihsk
ahikhs
ahiksh
ahiskh
ahishk
ahkihs
ahkish
ahkhis
```

Test Case: Any two (screenshot)



```
● shikhasingh@SHIKHAs-MacBook-Air C++ % cd "/Users/shikhasingh/Desktop/C++/" && g++ 13.permutation.cpp -o 13.per
  mutation && "/Users/shikhasingh/Desktop/C++/"13.permutation
  Enter a string: shikha
  ahhiks
  ahhisk
  ahhkis
  ahhksi
  ahhski
  ahhsik
  ahihks
  ahihsk
  ahikhs
  ahiksh
  ahiskh
  ahishk
  ahkihs
  ahkish
  ahkhis
  ahkhsi
  ahkshi
  ahksih
  ahsikh
  ahsihk
  ahskih
  ahskhi
  ahshki
  ahshik
  ahhiks
  ahhisk
  ahhkis
  ahhksi
  ahhski
  ahhsik
  ahihks
  ahihsk
  ahikhs
  ahiksh
  ahiskh
  ahishk
  ahkihs
  ahkish
  ahkhis
```



```
  p/C++/"13.permutation
● shikhasingh@SHIKHAs-MacBook-Air C++ % cd "/Users/shikhasingh/Desktop/C++"
● shikhasingh@SHIKHAs-MacBook-Air C++ % cd "/Users/shikhasingh/Desktop/C++/" && g++ 13.permutation.cpp -o 13.per
  mutation && "/Users/shikhasingh/Desktop/C++/"13.permutation
  Enter a string: mam
  amm
  amm
  mam
  mma
  mma
  mam
○ shikhasingh@SHIKHAs-MacBook-Air C++ % █
```

Conclusion:

The program uses recursion and backtracking to generate all possible permutations of a given string. It provides a fundamental understanding of recursion and string manipulation. Sorting the input string is done to ensure lexicographically ordered output. Users can input different strings to observe the program's behavior with various inputs.