

## **C++ PROGRAMMING LAB**



**Prepared by:**

Name of Student: **Mahajan Piyush Dnyaneshwar**

Roll No:**35**

Batch: **2023-27**

## **DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

<b>Exp. No</b>	<b>List of Experiment</b>
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 $1/1! + 2/2! + 3/3! + \dots + N/N!$ 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	<p>Create a C++ program to print the following pattern:</p> <pre>***** * * * * * * *****</pre>
15	<p>Write a C++ program to display the following pattern:</p> <pre>1 232 34543 4567654 34543 232</pre>
16	<p>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:</p> <ul style="list-style-type: none"> <li>• Create a <b>Product</b> class that represents a product in the inventory. Each <b>Product</b> object should have the following attributes: <ul style="list-style-type: none"> <li>• Product ID (an integer)</li> <li>• Product Name (a string)</li> <li>• Price (a floating-point number)</li> <li>• Quantity in stock (an integer)</li> </ul> </li> <li>• Implement a parameterized constructor for the <b>Product</b> class to initialize the attributes when a new product is added to the inventory.</li> </ul>
17	<p>Write a program to manage student records. Create a class <b>Student</b> 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.</p>
18	<p>Write a program that implements a basic calculator. Use a class <b>Calculator</b> 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.</p>

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:** Mahajan Piyush Dnyaneshwar

**Roll Number:** 35

### **Experiment No: 1**

---

**Title:** Write a program to find the roots of a quadratic equation.

**Theory:**

**Roots of a quadratic equation depends on the discriminant( $b^2-4ac$ ). If discriminant is**

**+ve, roots are real and positive. If it is -ve, roots are complex and different. If it is 0,**

**roots are real and same.**

**Code:**

```
#include <iostream>
```

```

#include <cmath>
using namespace std;

int main()
{
    float a, b, c, x1, x2, discriminant, realPart, imaginaryPart;
    cout << "Enter coefficients a, b and c: ";
    cin >> a >> b >> c;

    discriminant = b * b - 4 * a * c;

    if (discriminant > 0)
    {
        x1 = (-b + sqrt(discriminant)) / (2 * a);
        x2 = (-b - sqrt(discriminant)) / (2 * a);
        cout << "Roots are real and different." << endl;
        cout << "x1 = " << x1 << endl;
        cout << "x2 = " << x2 << endl;
    }

    else if (discriminant == 0)
    {
        cout << "Roots are real and same." << endl;
        x1 = -b / (2 * a);
        cout << "x1 = x2 =" << x1 << endl;
    }

    else
    {
        realPart = -b / (2 * a);
        imaginaryPart = sqrt(-discriminant) / (2 * a);
        cout << "Roots are complex and different." << endl;
    }

    return 0;
}

```

## **Output: (screenshot)**

```
● Enter coefficients a, b and c: 1 12 15
Roots are real and different.
x1 = -1.41742
x2 = -10.5826
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % █
```

## **Test Case: Any two (screenshot)**

```
● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p1.cpp -o p1 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p1
Enter coefficients a, b and c: 12 15 21
Roots are complex and different.
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % █
```

```
● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p1.cpp -o p1 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p1
Enter coefficients a, b and c: 21 1 14
Roots are complex and different.
```

## **Conclusion:**

**Hence, by checking the nature of discriminant(given by the user), we can find nature of roots and print them.**

## Experiment No: 2

**Title:** Write a program to calculate the power of a number using a loop.

**Theory:** Power of a number is calculated by multiplying the number by itself exponent times.

Eg-  $x^n = x \times x \times x \times \dots \times n$  times.  $2^3 = 2 \times 2 \times 2 = 8$ .

**Code:**

```
#include <iostream>
using namespace std;

int main(){
    int base, exp, i, result = 1;
    cout << "Enter base and exponent\n";
    cin >> base >> exp;

    for(i = 0; i < exp; i++){
        result = result * base;
    }

    cout << base << "^" << exp << " = " << result << endl;

    return 0;
}
```

## Output: (screenshot)

```
● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p2.cpp -o p2 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p2
Enter base and exponent
2 3
2^3 = 8
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
```

## Test Case: Any two (screenshot)

```
● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p2.cpp -o p2 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p2
Enter base and exponent
3 3
3^3 = 27
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
```

```
● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p2.cpp -o p2 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p2
Enter base and exponent
4 2
4^2 = 16
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
```

## Conclusion:

Hence, by using for loop to multiply the base by itself exponent times and printing the result to the user.

## Experiment No: 3

**Title:** Write a program to check if a given string, is a palindrome.

**Theory:**

A string is a palindrome if the string is the same when reversed. Eg- racecar is a palindrome as if it is reversed then it becomes racecar, hence both of them are equal.

Race is not a palindrome as when reversed it becomes scar, hence it is not equal.

**Code:**

```
#include<iostream>
#include<string>
using namespace std;
int main()
{
    string str,str1;
    cout<<"Enter a string: ";
    cin>>str;
    str1=str;
    int start=0,end=str.length()-1;
    while(start<end)
    {
        char temp=str[start];
        str[start]=str[end];
        str[end]=temp;
        start++;
        end--;
    }
    if (str1==str)
    {
        cout<<"String is a palindrome"<<endl;
    }
    else
    {
        cout<<"String is not a palindrome"<<endl;
    }
    return 0;
}
```

## Output: (screenshot)

```
● Enter a string: Piyush
String is not a palindrome
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
```

## Test Case: Any two (screenshot)

```
● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"tempCodeRunnerFile
Enter a string: mom
String is a palindrome
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
```

```
● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p3.cpp -o p3 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p3
Enter a string: mama
String is not a palindrome
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
```

## Conclusion:

Hence, by using while loop to reverse the string by swapping the characters from start to the end of string and comparing the reversed string to the original string and prints the appropriate message using if else statement.

## Experiment No: 4

**Title:** Write a program that simulates a simple ATM machine, allowing users to check their balance, deposit, or withdraw money using a switch statement.

**Theory:**

Using while loop and switch statement to print an ATM menu and take user's choice and perform the respective operation.

**Code:**

```
#include <iostream>
using namespace std;

int main() {

    float balance = 80000.0;
    int choice;
    float amount;

    while (true) {
        cout << "\t\t XYZ ATM \t\t" << endl;
        cout << "*****" << endl;
        cout << "1. Check Balance \t\t 2.Deposit Money\n" << endl;
        cout << "3. Withdraw Money \t\t 4.Exit" << endl;
        cout << "*****" << endl;
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
            case 1: [REDACTED]
                cout << "Your balance is: ₹" << balance <<
"\n" << endl; [REDACTED]
                break;

            //choice 2 to deposit money (only if amount to deposit
is >0) [REDACTED]
```

```
        case 2:
            cout << "Enter the amount to deposit: ₹";
            cin >> amount;
            if (amount > 0) {
                balance += amount;
                cout << "Deposited ₹" << amount << "
successfully.\n" << endl;
            } else {
                cout << "Invalid amount. Please enter a
positive amount.\n" << endl;
            }
            break;

        //choice 3 to withdraw money
        case 3:
            cout << "Enter the amount to withdraw: ₹";
            cin >> amount;

            if (amount > 0 && amount <= balance) {
                balance -= amount;
                cout << "Withdrawn ₹" << amount << "
successfully.\n" << endl;
            } else if (amount > balance) {
                cout << "Insufficient balance.\n" << endl;
            } else {
                cout << "Invalid amount. Please enter a
positive amount.\n" << endl;
            }
            break;

        case 4:
            cout << "Exiting the ATM. Have a nice day!" <<
endl;
    }

    default:
        cout << "Invalid choice. Please select a valid
option.\n" << endl;
        break;
    }
}

return 0;
}
```

## Output: (screenshot)

```
ush_20_C++_Lab_Practical/"p4
XYZ ATM
*****
1. Check Balance          2.Deposit Money
3. Withdraw Money         4.Exit
*****
Enter your choice: 1
Your balance is: ₹80000
```

## Test Case: Any two (screenshot)

```
XYZ ATM
*****
1. Check Balance          2.Deposit Money
3. Withdraw Money         4.Exit
*****
Enter your choice: 3
Enter the amount to withdraw: ₹21000
Withdrawn ₹21000 successfully.
```

```
XYZ ATM
*****
1. Check Balance          2.Deposit Money
3. Withdraw Money           4.Exit
*****
Enter your choice: 2
Enter the amount to deposit: ₹25000
Deposited ₹25000 successfully.
```

**Conclusion:** Hence, by using while loop to infinitely print ATM menu and using switch statement to perform user given operation(whether deposit or withdraw cash or check bank balance).

## **Experiment No: 5**

---

**Title: Write a program that finds the largest among three numbers using nested if-else statements.**

**Theory:**

**Using nested if else loop to first check whether num 1 is bigger than num 2, then if num 1 is bigger than num 3 or not. If num 2 is bigger than num 1, then check if num 2 is bigger than num 3 or not.**

**Code:**

```
#include <iostream>
using namespace std;

int main() {

    int num1, num2, num3;
    cout << "Enter first number: ";
    cin >> num1;

    cout << "Enter second number: ";
    cin >> num2;

    cout << "Enter third number: ";
    cin >> num3;

    if (num1 >= num2) {
        if (num1 >= num3) {
            cout << num1 << " is bigger than " << num2 << " and "
<< num3 << endl;
        } else {
            cout << num3 << " is bigger than " << num1 << " and "
<< num2 << endl;
        }
    } else {
        if (num2 >= num3) {
            cout << num2 << " is bigger than " << num1 << " and "
<< num3 << endl;
    }
}
```

```

        } else {
            cout << num3 << " is bigger than " << num1 << " and "
<< num2 << endl;
        }
    }
}

```

## Output: (screenshot)

```

cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p5.cpp -o p5 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p5
● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p5
Enter first number: 21
Enter second number: 12
Enter third number: 1
21 is bigger than 12 and 1
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %

```

## Test Case: Any two (screenshot)

```

● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p5
Enter first number: 12
Enter second number: 11
Enter third number: 5
12 is bigger than 11 and 5
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %

```

```

● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p5
Enter first number: 11
Enter second number: 16
Enter third number: 98
98 is bigger than 11 and 16
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %

```

**Conclusion: Hence, by using nested if else loop to check which number is the biggest among three numbers.**

## Experiment No: 6

**Title:** Write a program that determines the grade

**Theory:** Using if else-if ladder loop to check the marks given by user and printing the appropriate grade based on the marks. If  $\text{marks} \geq 90$ , grade-A.  $\text{Marks} \geq 80$ , grade-B.

**Marks** $\geq 70$ , grade-C.  $\text{Marks} \geq 60$ , grade-D. Else grade-F.

**Code:**

```
//Implement a program that determines the grade of a student based  
on their marks.
```

```
#include <iostream>  
using namespace std;  
  
int main() {  
    float marks;  
    cout << "Enter your marks: ";  
    cin >> marks; //prints grade depending on the marks given using  
if else if loop  
    if (marks >= 90) {  
        cout << "A";  
    } else if (marks >= 80) {  
        cout << "B";  
    } else if (marks >= 70) {  
        cout << "C";  
    } else if (marks >= 60) {  
        cout << "D";  
    } else {  
        cout << "F";  
    }  
    return 0;  
}
```

## **Output: (screenshot)**

```
● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p6.cpp -o p6 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p6
Enter your marks: 21
F2
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
```

## **Test Case: Any two (screenshot)**

```
● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p6.cpp -o p6 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p6
Enter your marks: 76
C6
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
```

```
● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p6.cpp -o p6 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p6
Enter your marks: 98
A8
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
```

## **Conclusion:**

**Hence, by using laddered if else-if loop, printing the grade based on the marks given by the user.**

## Experiment No: 7

### Title:

**Write a program to find the sum of digits of a number until it becomes a single-digit number.**

### Theory:

**Using while loop to find sum of digits of the number, then checking if the sum is single-digit or not. If not, then again using while loop to find sum of the previous sum of the digits. Then again checking if the new sum is single-digit or not and so on.**

### Code:

```
#include<iostream>
using namespace std;
int main()
{
    int n,r;
    cout<<"Enter a number: ";
    cin>>n;
    while (n>=10)
    {
        int sum=0;
        while(n>0)
        {
            r=n%10;
            sum+=r;
            n/=10;
        }
        n=sum;
    }
    cout<<"Sum of digits: "<<n<<endl;
    return 0;
}
```

## Output: (screenshot)

```
● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p7.cpp -o p7 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p7
Enter a number: 37
Sum of digits: 1
```

## Test Case: Any two (screenshot)

```
● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p7.cpp -o p7 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p7
Enter a number: 42
Sum of digits: 6
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % █
```

```
● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p7.cpp -o p7 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p7
Enter a number: 21
Sum of digits: 3
```

## Conclusion:

Hence, by using nested while loops, finding sum of digits of a number until the sum becomes a single digit number.

## Experiment No: 8

**Title: Write a program to print a Pascal's triangle using nested loops.**

**Theory:**

Pascal triangle is a triangular arrangement of numbers that gives the coefficients in the expansion of any binomial expression. The value of a number is calculated by the sum of two numbers above it.

**Code:**

```
#include <iostream>
using namespace std;

int main()
{
    int n;
    cout << "Enter number: ";
    cin >> n;

    while (n <= 0)
    {
        cout << "Invalid number" << endl;
        cout << "Enter number: " << endl;
        cin >> n;
    }

    cout << "The pattern is: " << endl
        << endl;

    for (int i = 1; i <= n; i++)
    {
        int num = 1;
        for (int j = 1; j <= n - i; j++)
        {
            cout << " ";
        }
        for (int k = 1; k <= i; k++)
```

```

    {
        cout << num << " ";
        num = num * (i - k) / k;
    }
    cout << endl;
}

return 0;
}

```

## Output: (screenshot)

```

s/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p8.cpp -o p8 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p8
Enter number: 4
The pattern is:

      1
     1 1
    1 2 1
   1 3 3 1
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %

```

## Test Case: Any two (screenshot)

```

● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p8.cpp -o p8 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p8
Enter number: 2
The pattern is:

      1
     1 1
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %

```

```
Enter number: 6
The pattern is:
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
 1 5 10 10 5 1
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
```

### Conclusion:

Hence, by using if else statement to check whether the user given number is positive or not and then using nested for loops(one for rows, other for printing whitespaces before the numbers, and another one for calculating the value to print) to print a pascal triangle of user given number of rows.

## Experiment No: 9

**Title:** Write a program to calculate the sum of series  $1/1! + 2/2! + 3/3! + \dots + N/N!$  using nested loops.

**Theory:**

Using factorial function to calculate factorial of a number. Taking range from the user and using a for loop to calculate sum of the series using the factorial function to calculate the denominator and printing the sum at the end.

**Code:**

```
#include <iostream>
using namespace std;

int fact(int n) {
    return (n > 0) ? n * fact(n - 1) : 1;
}

int main() {

    int num;
    double sum = 0.0;

    cout << "Enter number: ";
    cin >> num;

    cout << "The sum of the following series" << endl;

    for (int i = 1; i <= num; i++) {
        int nFact = 1;
        for (int j = 1; j <= i; j++) {
            nFact = nFact * j;
        }

        if (i < num) {
            cout << i << "/" << i << "! + ";
        }
    }
}
```

```

        } else {
            cout << i << "/" << i << " ! = ";
        }

        sum += double(i) / nFact;
    }

    cout << sum << endl;

    return 0;
}

```

## Output: (screenshot)

```

● Enter number: 6
The sum of the following series
1/1! + 2/2! + 3/3! + 4/4! + 5/5! + 6/6! = 2.71667
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %

```

## Test Case: Any two (screenshot)

```

● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ program9.cpp -o program9 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"program9
Enter number: 4
The sum of the following series
1/1! + 2/2! + 3/3! + 4/4! = 2.66667
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %

● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ program9.cpp -o program9 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"program9
Enter number: 7
The sum of the following series
1/1! + 2/2! + 3/3! + 4/4! + 5/5! + 6/6! + 7/7! = 2.71806
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %

```

## **Conclusion:**

Hence, by using nested for loops and recursive factorial function to calculate sum of he series till the range given by the user and printing the sum.

## **Experiment No: 10**

---

**Title: Write a program to create an array of strings and display them in alphabetical order.**

### **Theory:**

**Creating an array of strings of size as specified by the user. Using bubble sort algorithm to sort the strings inside the array in alphabetical order and printing the array to the user.**

### **Code:**

```
#include<iostream>
#include<string>
using namespace std;
int main()
{
    int n;
    cout<<"Enter terms of string: ";
    cin>>n;
    string str[n];
    cout<<"Enter "<<n<<" strings: "<<endl;
    for (int i=0; i<n; i++)
    {
        cin>>str[i];
    }
    cout<<endl<<"String: "<<endl;
```

```

for (int i=0; i<n; i++)
{
    cout<<str[i]<<endl;
}
for (int i=0; i<n; i++)
{
    for (int j=0; j<n-1; j++)
    {
        if (str[j]>str[j+1])
        {
            string temp=str[j];
            str[j]=str[j+1];
            str[j+1]=temp;
        }
    }
}
cout<<endl<<"In alphabetical order: "<<endl;
for (int i=0; i<n; i++)
{
    cout<<str[i]<<endl;
}
return 0;
}

```

## Output: (screenshot)

```

mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p10.cpp -o p10 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p10
Enter terms of string: 3
Enter 3 strings:
c
a
b

String:
c
a
b

In alphabetical order:
a
b
c
mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %

```

## Test Case: Any two (Screenshot)

```
● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p10.cpp -o p10 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p10
Enter terms of string: 2
Enter 2 strings:
qqe
a sa

String:
qqe
a

In alphabetical order:
a
qqe
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
```

```
● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p10.cpp -o p10 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p10
Enter terms of string: 2
Enter 2 strings:
z
q

String:
z
q

In alphabetical order:
q
z
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
```

## Conclusion:

Hence, by using nested for loops and sorting the strings in the array in alphabetical order and printing the array.

## Experiment No: 11

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

**Theory:**

Creating an array of integers of size given by the user and adding the numbers from he user and using for loop to check if the array is sorted in ascending order or not and printing the appropriate message using if else statements and a flag(sorted).

**Code:**

```
#include<iostream>
using namespace std;
int main()
{
    int n;bool sorted = true;
    cout<<"Enter length of array: ";
    cin>>n;
    int arr[n];
    cout<<"Enter "<<n<<" elements: "<<endl;
    for (int i=0; i<n; i++)
    {
        cin>>arr[i];
    }
    cout<<"Array: |";
    for (int i=0; i<n; i++)
    {
        cout<<arr[i]<<",";
    }
    cout<<"|"<<endl;
    for (int i=0; i<n-1; i++)
    {
        if (arr[i]>arr[i+1])
        {
            sorted = false;
            break;
        }
    }
    if (sorted == true)
    {
        cout<<"Array is sorted in ascending order"<<endl;
    }
    else
```

```
{  
    cout<<"Array is not sorted in ascending order"=<endl;  
}  
return 0;  
}
```

## Output: (screenshot)

```
● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p11.cpp -o p11 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p11  
Enter length of array: 3  
Enter 3 elements:  
1 5 8  
Array: |1,5,8,|  
Array is sorted in ascending order  
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
```

## Test Case: Any two (screenshot)

```
● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p11.cpp -o p11 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p11  
Enter length of array: 5  
Enter 5 elements:  
1 17 2 0 21  
Array: |1,17,2,0,21,|  
Array is not sorted in ascending order  
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
```

```
● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p11.cpp -o p11 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p11  
Enter length of array: 3  
Enter 3 elements:  
9 17 64  
Array: |9,17,64,|  
Array is sorted in ascending order  
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
```

## Conclusion:

Hence, by using for loop to check if the array of numbers is sorted in ascending order and updating the value of flag(sorted) accordingly and printing the appropriate message using if else statements and the value of flag.

## Experiment No: 12

**Title: Write a program to calculate the sum of elements in each row of a matrix.**

### Theory:

Creating a 2d array of row and column given by the user and filling them with values given by the user and using nested for loop to calculate sum of elements in each row of the array and printing it.

### Code:

```
#include <iostream>
using namespace std;

int main() {
    int rows, col;
    cout << "Enter row and colum number: " << endl;
    cin >> rows >> col;

    int arr[rows][col];
    for(int i = 0; i < rows; i++){
        for(int j = 0; j < col; j++){
            cout << "Enter value of arr[" << i << "] [" << j << "]: ";
            cin >> arr[i][j];
        }
    }
    cout << endl;
```

```

    }
}

int sum=0;
for(int i=0; i < rows; i++) {
    for (int j = 0; j < col; j++) {
        sum+=arr[i][j];
    }
    cout <<"Sum of row " <<i << " is:"<<sum << endl;
    sum=0;
}
}

```

## Output: (screenshot)

```

Enter value of arr[0][3]: 24
Enter value of arr[0][4]: 54
Enter value of arr[0][5]: 131

Enter value of arr[1][0]: clear
Enter value of arr[1][1]: Enter value of arr[1][2]: Enter value of arr[1][3]: Enter value of arr[1][4]: En
ter value of arr[1][5]:
Sum of row 0 is:1756
Sum of row 1 is:1870753791

```

## Test Case: Any two (screenshot)

```

● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Download
s/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p12.cpp -o p12 && "/Users/mahajanpiyush/Downloads/Mahajan_P
iyush_20_C++_Lab_Practical/"p12
Enter row and colum number:
2 3
Enter value of arr[0][0]: 12
Enter value of arr[0][1]: 12
Enter value of arr[0][2]: 54

Enter value of arr[1][0]: 123
Enter value of arr[1][1]: 12
Enter value of arr[1][2]: 54

Sum of row 0 is:78
Sum of row 1 is:189
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %

```

```
Enter value of arr[0][0]: 1
Enter value of arr[0][1]: 2

Enter value of arr[1][0]: 3
Enter value of arr[1][1]: 4

Sum of row 0 is:3
Sum of row 1 is:7
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
```

## Conclusion:

Hence, by using nested for loops(one for rows, other for column), calculating the sum of elements in each row of the matrix and printing it.

## Experiment No: 13

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

### Theory:

Make a factorial function to calculate factorial of a number(since there are as many permutations of a string as the factorial of number of characters in the string). Then making a permutation function to generate permutations of the string by fixing one character and swapping other characters and moving from left to right in the string and doing the same procedure again till the pointer reaches to the right corner of the string.

### Code:

```
#include <iostream>
#include <string>
using namespace std;
```

```
int fact(int a)
{
    if (a == 0 || a == 1)
```

```
{  
    return 1;  
}  
else  
{  
    return a * fact(a - 1);  
}  
}
```

```
void swap(char &a, char &b)  
{  
    char temp = a;  
    a = b;  
    b = temp;  
}  
  
int isFound(string* arr, string query, int n) {  
    for (int i = 0; i < n; i++) {  
        if (arr[i] == query) {  
            return 1;  
        }  
    }  
    return 0;  
}
```

```
void generatePermutations(string str, int left, int right, int  
&count, string* words)  
{  
  
    if (left == right)  
    {  
        if(!isFound(words, str, count)) {  
            cout << count+1 << ":" << str << endl;  
            words[count] = str;  
            count++;  
        }  
        return;  
    }  
    else  
    {  
        for (int i = left; i < right; ++i)  
        {  
  
            swap(str[left], str[i]);  
            generatePermutations(str, left + 1, right, count,  
words);  
        }  
    }  
}
```

```
        swap(str[left], str[i]);
    }
}
}

int main()
{
    string input;
    cout << "Enter a string: ";
    cin >> input;
    int n = input.length();
    cout << "Length of string: " << n << endl;
    int length = fact(n);
    cout << "Number of permutations: " << length << endl;
    cout << "All permutations of the string are:" << endl;
    int count = 0;
    string permutations[length];
    generatePermutations(input, 0, n, count, permutations);
    return 0;
}
```

## Output: (screenshot)

```
Length of string: 3
Number of permutations: 6
All permutations of the string are:
1: mom
2: mmo
3: omm
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
```

## Test Case: Any two (screenshot)

```
Enter a string: rj
Length of string: 2
Number of permutations: 2
All permutations of the string are:
1: rj
2: jr
mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
```

```
mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p13.cpp -o p13 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p13
Enter a string: bye
Length of string: 3
Number of permutations: 6
All permutations of the string are:
1: bye
2: bey
3: ybe
4: yeb
5: eyb
6: eby
mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
```

## Conclusion:

Hence, by using factorial and permutation function to calculate all possible permutations of a string given by the user.

## Experiment No: 14

**Title: Create a C++ program to print the following pattern:**

```
*****
* *
* *
* *
*****

```

## Theory:

Using nested for loops to print rows and columns and using if else statement to print stars in specific rows and columns.

## Code:

```
#include <iostream>
using namespace std;

int main() {

    int num;
    cout << "Enter the number of the lines: ";
    cin >> num;

    cout << endl << "The pattern with " << num << " rows is" <<
    endl << endl;

    for (int i = 0; i < num; i++) {
        for (int j = 0; j < num; j++) {
            if (i == 0 || i == num - 1 || j == 0 || j == num - 2) {
                cout << "*";
            } else {
                cout << " ";
            }
        }
        cout << endl;
    }

    return 0;
}
```

## Output: (screenshot)

```
iyush_20_C++_Lab_Practical/"p14
Enter the number of the lines: 3

The pattern with 3 rows is

***  
**  
***  
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p14.cpp -o p14 && "/Users/mahajanpiyush/Downloads/Mahajan_P
```

## Test Case: Any two (screenshot)

```
● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p14.cpp -o p14 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p14
Enter the number of the lines: 4

The pattern with 4 rows is

****  
* *  
* *  
****  
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
```

```
****  
* *  
* *  
****  
● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p14.cpp -o p14 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p14
Enter the number of the lines: 6

The pattern with 6 rows is

*****  
* *  
* *  
* *  
* *  
*****  
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
```

## Conclusion:

**Hence, by using nested for loops and if else statements to print stars in specific rows and columns to print a pattern.**

## **Experiment No: 15**

---

**Title: Write a C++ program to display the following pattern:**

```
1  
232  
34543  
4567654  
34543  
232
```

**Theory:**

**Using nested for loops to print rows and columns of upper half and another nested for loop to print rows and columns of lower half.**

**Code:**

```
#include <iostream>  
using namespace std;  
  
int main() {  
  
    int num;  
    cout << "Enter the number of lines: ";  
    cin >> num;
```

```
    cout << endl << "The pattern with " << num << " rows is" <<
endl << endl ;
```

```
    for (int i = 1; i <= num; i++) {
        for (int j = 1; j <= num - i; j++) {
            cout << " ";
        }
```

```
        for (int k = i; k <= 2 * i - 1; k++) {
            cout << k;
        }
```

```
        for (int l = 2 * i - 2; l >= i; l--) {
            cout << l;
        }
```

```
        cout << endl;
    }
```

```
    for (int i = num - 1; i >= 1; i--) {
        for (int j = 1; j <= num - i; j++) {
            cout << " ";
        }
```

```
        for (int k = i; k <= 2 * i - 1; k++) {
            cout << k;
        }
```

```
        for (int l = 2 * i - 2; l >= i; l--) {
            cout << l;
        }
```

```
        cout << endl;
    }
```

```
    return 0;
}
```

## Output: (screenshot)

```
● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p15.cpp -o p15 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p15
Enter the number of lines: 3
```

The pattern with 3 rows is

```
1
232
34543
232
1
```

```
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % █
```

## Test Case: Any two (screenshot)

```
232
1
● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p15.cpp -o p15 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p15
Enter the number of lines: 5
```

The pattern with 5 rows is

```
1
232
34543
4567654
567898765
4567654
34543
232
1
```

```
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % █
```

```
● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p15.cpp -o p15 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p15
Enter the number of lines: 2
```

The pattern with 2 rows is

```
1
232
1
```

```
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % █
```

## **Conclusion:**

**Hence, by using nested for loops(for rows and columns, for printing whitespaces before numbers, and for the numbers itself) for printing upper and lower half of the pyramid of user given number of rows.**

## **Experiment No: 16**

---

### **Title:**

**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.**

## Theory:

A class is a blueprint for objects. It consists of attributes and methods. An object is an instance of a class. It has a copy of the attributes and shares the methods with other objects. A parameterised constructor is used to initialise the attributes when an object is created with some arguments.

## Code:

```
#include <iostream>
#include <string>
using namespace std;

class Product
{
private:
    int prod_id;
    string prod_name;
    float price;
    int quantity;

public:
    Product()
    {
    }
    Product(int id, string n, float p, int q)
    {
        prod_id = id;
        prod_name = n;
        price = p;
        quantity = q;
    }
};

int main()
{
    int n, prod_id, quantity;
    string prod_name;
    float price;
    cout << "Enter number of products: ";
    cin >> n;
    Product p[n];
    for (int i = 0; i < n; i++)
    {
        cout << "Enter product details: ";
        cout << "Product ID: ";
        cout << prod_id;
        cout << "Product Name: ";
        cout << prod_name;
        cout << "Price: ";
        cout << price;
        cout << "Quantity: ";
        cout << quantity;
        cout << endl;
    }
}
```

```
{  
    cout << "Enter Product ID: ";  
    cin >> prod_id; [REDACTED]  
    cout << "Enter Product Name: ";  
    cin.ignore(); [REDACTED]  
    getline(cin, prod_name); [REDACTED]  
    cout << "Enter Price of Product: ";  
    cin >> price; [REDACTED]  
    cout << "Enter Quantity of Product: ";  
    cin >> quantity; [REDACTED]  
    p[i] = Product(prod_id, prod_name, price, quantity);  
}  
return 0;  
}
```

## Output: (screenshot)

```
● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p16.cpp -o p16 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p16  
Enter number of products: 2  
Enter Product ID: 1212  
Enter Product Name: Apple  
Enter Price of Product: 21  
Enter Quantity of Product: 1  
Enter Product ID: 9812  
Enter Product Name: Banana  
Enter Price of Product: 213  
Enter Quantity of Product: 3  
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
```

## Test Case: Any two (screenshot)

```
● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p16.cpp -o p16 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p16
Enter number of products: 2
Enter Product ID: 13213
Enter Product Name: iphone
Enter Price of Product: 12413231
Enter Quantity of Product: 90
Enter Product ID: 1241314
Enter Product Name: Yeppe
Enter Price of Product: 31
Enter Quantity of Product: 12
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
```

```
● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p16.cpp -o p16 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p16
Enter number of products: 1
Enter Product ID: 1313
Enter Product Name: phone
Enter Price of Product: 2133
Enter Quantity of Product: 30
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
```

## Conclusion:

Hence, by using classes and list of objects of size given by the user and using a parameterised constructor to initialise

## Experiment No: 17

**Title: 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.**

## Theory:

Using parameterised constructor to show details of students. Using for loop to take student details from the user and calculating average of students.

## Code:

```
#include <iostream>
#include <string>
using namespace std;

class Student
{
private:
    string name;
    int roll;
    float marks;

public:
    Student() {}
    Student(string n, int r, float m)
    {
        name = n;
        roll = r;
        marks = m;
    }

    void getData()
    {
        cout << endl
            << "Name of student: " << name << endl;
        cout << "Roll no of student: " << roll << endl;
        cout << "Marks of student: " << marks << endl;
    }
};

int main()
{
    int n, roll;
    float marks, sum=0;
    string name;
    cout << "Enter number of students: ";
    cin >> n;
    Student s[n];
    for (int i = 0; i < n; i++)
}
```

```
{  
    cout << "Enter name of student: ";  
    cin.ignore();  
    getline(cin, name);  
    cout << "Enter roll number of student: ";  
    cin >> roll;  
abc:  
    cout << "Enter marks of student(out of 100): ";  
    cin >> marks;  
    if (marks > 100)  
    {  
        goto abc;  
    }  
    sum += marks;  
    s[i] = Student(name, roll, marks);  
}  
double avg = sum / n;  
for (int i = 0; i < n; i++)  
{  
    s[i].getData();  
}  
cout << endl  
    << "Sum of marks: " << sum << endl;  
cout << "Average of marks: " << avg << endl;  
return 0;  
}
```

**Output: (screenshot)**

```
● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p17.cpp -o p17 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p17
Enter number of students: 2
Enter name of student: Piyush
Enter roll number of student: 21
Enter marks of student(out of 100): 13
Enter name of student: Yash
Enter roll number of student: 32
Enter marks of student(out of 100): 78

Name of student: Piyush
Roll no of student: 21
Marks of student: 13

Name of student: Yash
Roll no of student: 32
Marks of student: 78
● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p17.cpp -o p17 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p17
Enter number of students: 2
Enter name of student: Piyush
Enter roll number of student: 21
Enter marks of student(out of 100): 13
Enter name of student: Yash
Enter roll number of student: 32
Enter marks of student(out of 100): 78

Name of student: Piyush
Roll no of student: 21
Marks of student: 13

Name of student: Yash
Roll no of student: 32
Marks of student: 78

Sum of marks: 91
Average of marks: 45.5
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
```

## Test Case: Any two (screenshot)

```
● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p17.cpp -o p17 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p17
Enter number of students: 2
Enter name of student: Piyush
Enter roll number of student: 21
Enter marks of student(out of 100): 13
Enter name of student: Yash
Enter roll number of student: 32
Enter marks of student(out of 100): 78

Name of student: Piyush
Roll no of student: 21
Marks of student: 13

Name of student: Yash
Roll no of student: 32
Marks of student: 78

Sum of marks: 91
Average of marks: 45.5
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
```

```
● mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p17.cpp -o p17 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p17
Enter number of students: 1
Enter name of student: Kiya
Enter roll number of student: 21
Enter marks of student(out of 100): 56

Name of student: Kiya
Roll no of student: 21
Marks of student: 56

Sum of marks: 56
Average of marks: 56
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
```

### **Conclusion:**

Hence, by using parameterised constructor and for loop to get student details and print them and calculating and printing the average of all the students given by the user.

### **Experiment No: 18**

---

**Title: 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.**

**Theory: Creating a class and making a method for getting values from the user and printing a menu of different operations to perform and performing the user given operation on the values and printing the result using while loop.**

### **Code:**

```
#include <iostream>
using namespace std;
```

```
class Calculator
{
private:
    int a;
    float b, c;

public:
    void calculate()
    {
        while (true)
        {
            cout << "Enter first number: ";
            cin >> b;
            cout << "Enter second number: ";
            cin >> c;
            cout << "Calculator: " << endl
                << "Press " << endl
                << "1 for Addition" << endl
                << "2 for Subtraction" << endl
                << "3 for Multiplication" << endl
                << "4 for Division" << endl
                << "0 to end" << endl;
            cin >> a;
            switch (a)
            {
                case 1:
                    addition(b, c);
                    break;
                case 2:
                    subtraction(b, c);
                    break;
                case 3:
                    multiplication(b, c);
                    break;
                case 4:
                    division(b, c);
                    break;
                case 0:
                    return;
                default:
                    cout << "Invalid choice! Please enter a valid
choice" << endl;
            }
        }
    }

    void addition(float x, float y)
    {
        cout << "Addition: " << x + y << endl;
    }

    void subtraction(float x, float y)
    {
        cout << "Subtraction: " << x - y << endl;
    }

    void multiplication(float x, float y)
    {
        cout << "Multiplication: " << x * y << endl;
    }

    void division(float x, float y)
    {
        cout << "Division: " << x / y << endl;
    }
}
```

```

    }
    void subtraction(float x, float y)
    {
        cout << "Subtraction: " << x - y << endl;
    }
    void multiplication(float x, float y)
    {
        cout << "Multiplication: " << x * y << endl;
    }
    float division(float x, float y)
    {
        if (x == 0 || y == 0)
        {
            cout << "Invalid number" << endl;
            return 0;
        }
        else
        {
            cout << "Division: " << x / y << endl;
            return 0;
        }
    }
};

int main()
{
    Calculator obj;
    obj.calculate();
    return 0;
}

```

## Output: (screenshot)

```

○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p18.cpp -o p18 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p18
Enter first number: 21
Enter second number: 2
Calculator:
Press
1 for Addition
2 for Subtraction
3 for Multiplication
4 for Division
0 to end
1
Addition: 23
Enter first number: █

```

## Test Case: Any two (screenshot)

```
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p18.cpp -o p18 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p18
Enter first number: 21
Enter second number: 43
Calculator:
Press
1 for Addition
2 for Subtraction
3 for Multiplication
4 for Division
0 to end
2
Subtraction: -22
Enter first number: █
```

```
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical % cd "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/" && g++ p18.cpp -o p18 && "/Users/mahajanpiyush/Downloads/Mahajan_Piyush_20_C++_Lab_Practical/"p18
Enter first number: 12
Enter second number: 98
Calculator:
Press
1 for Addition
2 for Subtraction
3 for Multiplication
4 for Division
0 to end
4
Division: 0.122449
Enter first number: █
```

## Conclusion:

Hence, by using while loop in a method for printing the menu of operations and creating their methods and calling the user given operation method and printing the result to the user.

## Experiment No: 19

**Title: 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.**

**Theory: Creating a class Product and creating methods for adding product name, price, and quantity to the cart and displaying the cart at the end.**

**Code:**

```
#include <iostream>
using namespace std;

class Product
{
private:
    string name, prod[5];
    float prices[5], sum = 0;
    int quantity, quan[5], n;

public:
    Product()
    {
        cout << "Enter number of products: ";
        cin >> n;
        prod[n];
        prices[n];
        quan[n];
        for (int i = 0; i < n; i++)
        {
            cout << "Enter name of product: ";
            cin.ignore();
            getline(cin, name);
            prod[i] = name;
            cout << "Enter cost: ";
            cin >> prices[i];
            // prices[i]=price;
            cout << "Enter quantity: ";
            cin >> quantity;
            quan[i] = quantity;
            sum += (prices[i] * quan[i]);
        }
    }
}
```

```

        }
    }

void cart()
{
    cout << "Cart: " << endl
        << "Product Name"
        << "\t"
        << "Price"
        << "\t"
        << "Quantity"
        << "\t" << endl;
    for (int i = 0; i < n; i++)
    {
        cout << prod[i] << "\t\t" << prices[i] << "\t" <<
        quan[i] << endl;
    }
    cout << "Total cost: " << sum << endl;
}
};

int main()
{
    Product p1;
    p1.cart();
    return 0;
}

```

## Output: (screenshot)

```

4 warnings generated.
Enter number of products: 1
Enter name of product: Phone
Enter cost: 21000
Enter quantity: 1
Cart:
Product Name      Price      Quantity
Phone            21000      1
Total cost: 21000

```

## Test Case: Any two (screenshot)

```
Enter number of products: 2
Enter name of product: phobe
Enter cost: 12
Enter quantity: 1
Enter name of product: lolipop
Enter cost: 2100
Enter quantity: 12
Cart:
Product Name      Price      Quantity
phobe            12          1
lolipop          2100        12
Total cost: 25212
```

```
Enter number of products: 1
Enter name of product: kella
Enter cost: 21
Enter quantity: 43
Cart:
Product Name      Price      Quantity
kella            21          43
Total cost: 903
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
```

**Conclusion:** Hence, by using for loop to ask for product details by the user for user given number of products and displaying the cart at the end with total cost and product details.

## Experiment No: 20

**Title:** 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 initialise and release resources.

### Theory:

Using constructor to get name of student and methods to get grades of the student and store in an array and using for loop to calculate sum of grades and display average grade and student name.

### Code:

```
#include <iostream>
#include <string>
using namespace std;

class Student
{
private:
    string name;
    char grades, grade[5];
    int sum = 0, avg = 0, n, marks[5];

public:
    Student()
    {
        cout << "Enter name of student: ";
        getline(cin, name);
    }
    void addGrade()
    {
        cout << "Enter number of subjects: ";
        cin >> n;
```

```

        for (int i = 0; i < n; i++)
    {
        cout << "Enter " << i + 1 << " subject's "
grade(A,B,C,D,E,F): ";
        cin >> grade[i];
    }
}
void averageGrade()
{
    for (int i = 0; i < n; i++)
    {
        if (tolower(grade[i]) == 'a')
        {
            marks[i] = 100;
        }
        else if (tolower(grade[i]) == 'b')
        {
            marks[i] = 90;
        }
        else if (tolower(grade[i]) == 'c')
        {
            marks[i] = 80;
        }
        else if (tolower(grade[i]) == 'd')
        {
            marks[i] = 70;
        }
        else if (tolower(grade[i]) == 'e')
        {
            marks[i] = 60;
        }
        else
        {
            marks[i] = 50;
        }
    }
    for (int i = 0; i < n; i++)
    {
        sum += marks[i];
    }
    avg = sum / n;
    if (avg > 90)
    {
        grades = 'A';
    }
    else if (avg > 80 && avg <= 90)
    {
        grades = 'B';
    }
}

```

```

        else if (avg > 70 && avg <= 80)
    {
        grades = 'C';
    }
    else if (avg > 60 && avg <= 70)
    {
        grades = 'D';
    }
    else if (avg > 50 && avg <= 60)
    {
        grades = 'E';
    }
    else
    {
        grades = 'F';
    }
}
void showDetails()
{
    cout << endl
        << "Name of Student: " << name << endl;
    cout << "Grades: ";
    for (int i = 0; i < n; i++)
    {
        cout << (char)toupper(grade[i]) << " ";
    }
    cout << endl
        << "Average grade: " << grades << endl;
}
~Student()
{
    cout << "Destructor is called." << endl;
}
};

int main()
{
    Student s1;
    s1.addGrade();
    s1.averageGrade();
    s1.showDetails();
    return 0;
}

```

## **Output: (screenshot)**

```
Enter name of student: Piyush
Enter number of subjects: maths

Name of Student: Piyush
Grades:
Average grade: F
Destructor is called.
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
```

## **Test Case: Any two (screenshot)**

```
Enter name of student: Lol
Enter number of subjects: English

Name of Student: Lol
Grades:
Average grade: F
Destructor is called.
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
```

```
Enter name of student: piyush
Enter number of subjects: 1
Enter 1 subject's grade(A,B,C,D,E,F): A

Name of Student: piyush
Grades: A
Average grade: A
Destructor is called.
○ mahajanpiyush@Mahajans-MacBook-Air Mahajan_Piyush_20_C++_Lab_Practical %
```

## **Conclusion:**

**Hence, by using constructors and destructors to get student name and methods to get student grades and calculating average grade using for loop and printing student details and average grade to the user.**