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

C++ PROGRAMMING LAB

Prepared by:

Name of Student: __Sparsh Sharma____

Roll No: ___37________

Batch: 2023-27

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Exp. No	**List of Experiment**
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: __Sparsh Sharma____

Roll Number: _____37_______

Experiment No: 1

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

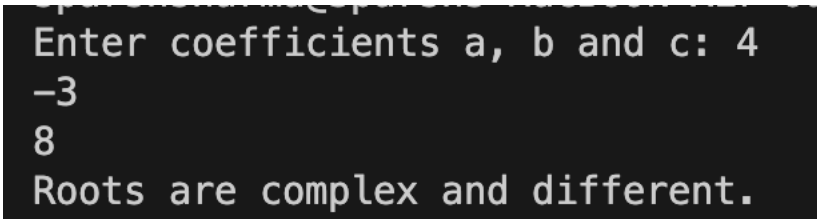
Theory: **Roots of a quadratic equation depends on the discriminant(b^2-4ac). If discriminant is +ve, roots are real and positive. If it is -ve, roots are complex and different. If it is 0, roots are real and same.**

Code:

// to find the roots of a quadratic equation

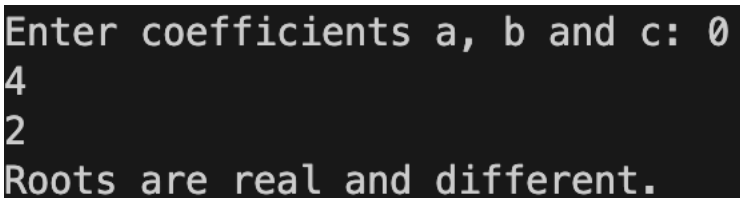
```
#include <iostream>
#include <cmath>
using namespace std;

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: 4
-3
8
Roots are complex and different.
```

Test Case: Any two (screenshot)



```
Enter coefficients a, b and c: 0
4
2
Roots are real and different.
```

Conclusion:

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

Name of Student: __Sparsh Sharma____

Roll Number: _____37_______

Experiment No: 2

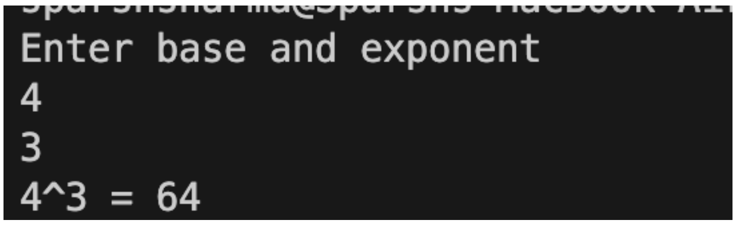
Title: Write a program to calculate the power of a number using a loop.

Theory:Power of a number is calculated by multiplying the number by itself exponent times. Eg- x^n=x*x*x*x…n times. 2^3=2*2*2=8.

Code:

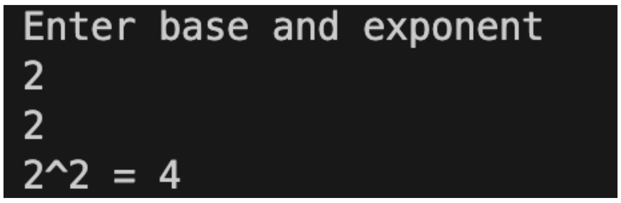
```
//program to calculate power of a number using loop
#include <iostream>

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)

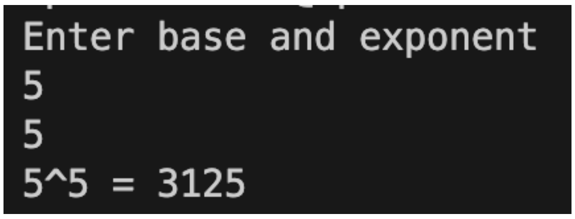


```
[illegible]
Enter base and exponent
4
3
4^3 = 64
```

Test Case: Any two (screenshot)



```
Enter base and exponent
2
2
2^2 = 4
```



```
Enter base and exponent
5
5
5^5 = 3125
```

Conclusion:

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

Name of Student: __Sparsh Sharma____

Roll Number: _____37_______

Experiment No: 3

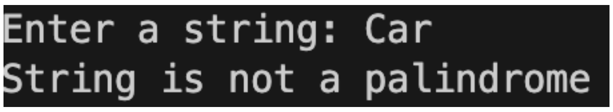
Title: Write a program to check if a given string, is a palindrome.

Theory: A string is a palindrome if the string is the same when reversed. Eg- race car is a palindrome as if it is reversed then it becomes race-car, hence both of them are equal. Race is not a palindrome as when reversed it becomes scar, hence it is not equal.

Code:

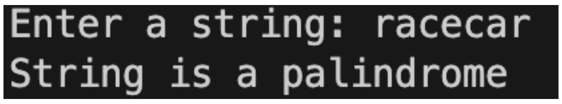
```
//program to check whether a given string is a palindrome
#include<iostream>
#include<string>
using namespace std;
int main()
{
    string str,str1;
    cout<<"Enter a string: ";
    cin>>str;
    str1=str;
    int start=0,end=str.length()-1;
    while(start<end)
    {
        char temp=str[start];
        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)

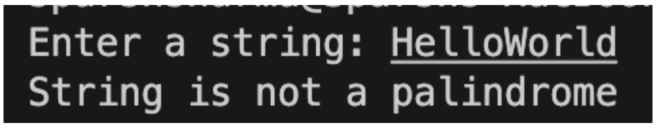


```
Enter a string: Car
String is not a palindrome
```

Test Case: Any two (screenshot)



```
Enter a string: racecar
String is a palindrome
```



```
Enter a string: HelloWorld
String is not a palindrome
```

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.

Name of Student: __Sparsh Sharma____

Roll Number: _____37_______

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:

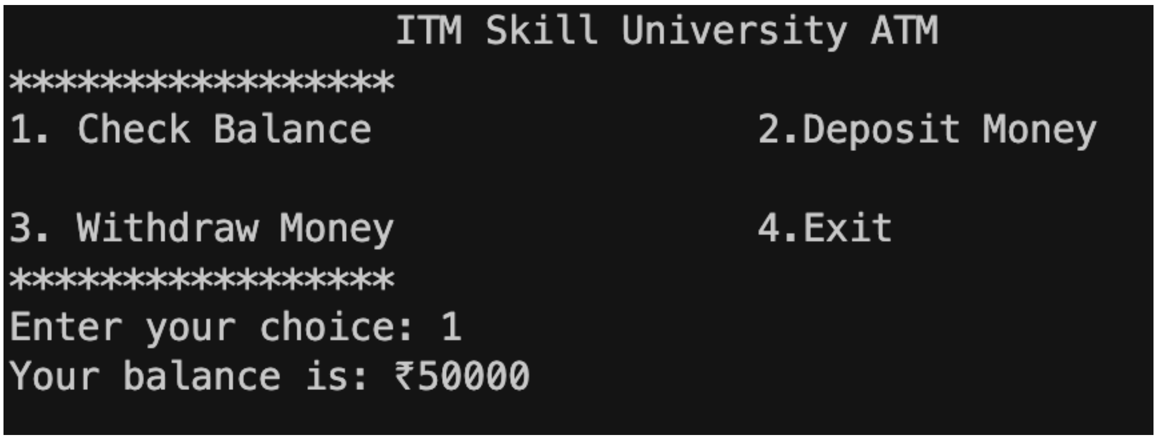
Using while loop and switch statement to print an ATM menu and take user's choice and perform the respective operation.

Code:

//Implement a C++ program that simulates a simple ATM machine, allowing users to check their balance, deposit, or withdraw money using a switch statement.

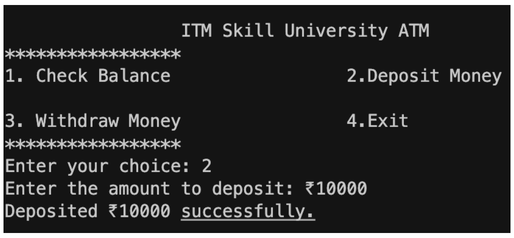
```
#include <iostream>
using namespace std;

int main() {

    float balance = 50000.0;
    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:
                cout << "Your balance is: ₹" << balance <<
"\n"<<endl;
                break;
            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;

            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;
                return 0;
            default:
                cout << "Invalid choice. Please select a valid 
option.\n" << endl;
                break;
        }
    }

    return 0;
}
```

Output: (screenshot)

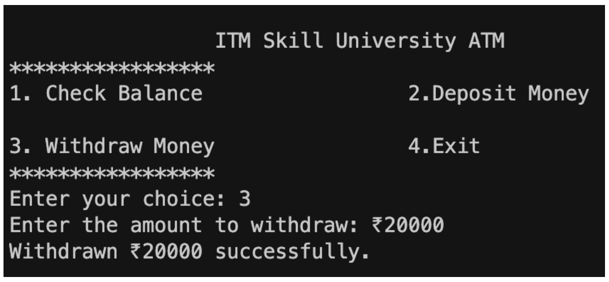


```
                 ITM Skill University ATM
*****************
1. Check Balance                 2.Deposit Money

3. Withdraw Money                4.Exit
*****************
Enter your choice: 1
Your balance is: ₹50000
```

Test Case: Any two (screenshot) :-



```
                 ITM Skill University ATM
*****************
1. Check Balance                 2.Deposit Money

3. Withdraw Money                4.Exit
*****************
Enter your choice: 2
Enter the amount to deposit: ₹10000
Deposited ₹10000 successfully.
```



```
                 ITM Skill University ATM
*****************
1. Check Balance                 2.Deposit Money

3. Withdraw Money                4.Exit
*****************
Enter your choice: 3
Enter the amount to withdraw: ₹20000
Withdrawn ₹20000 successfully.
```

Conclusion:

The Conclusion is 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.

Name of Student: __Sparsh Sharma____

Roll Number: ______37_______

Experiment No: 5

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

Theory:

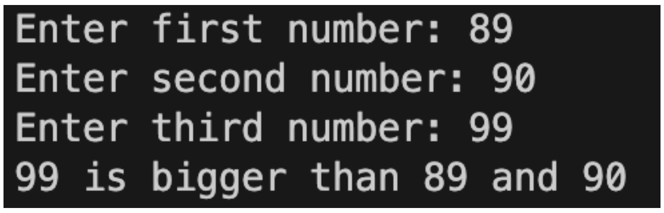
Using nested if else loop to first check whether num 1 is bigger than num 2, then if num 1 is bigger than num 3 or not. If num 2 is bigger than num 1, then check if num 2 is bigger than num 3 or not.

Code:

```
// to find largest among three numbers using nested if else loop
#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;
```

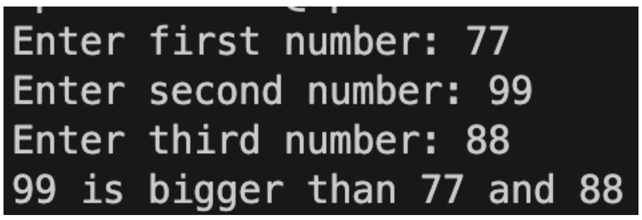
```
            }
        } else {
            if (num2 >= num3) {
                cout << num2 << " is bigger than " << num1 << " and "
<< num3 << endl;
            } else {
                cout << num3 << " is bigger than " << num1 << " and "
<< num2 << endl;
            }
        }
}
```

Output: (screenshot)

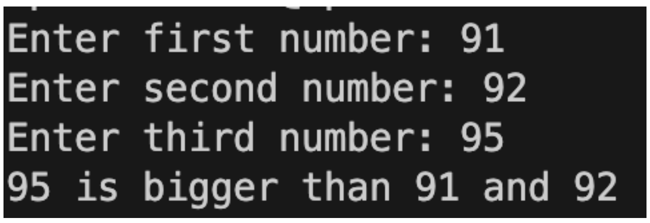


```
Enter first number: 89
Enter second number: 90
Enter third number: 99
99 is bigger than 89 and 90
```

Test Case: Any two (screenshot)



```
Enter first number: 77
Enter second number: 99
Enter third number: 88
99 is bigger than 77 and 88
```



```
Enter first number: 91
Enter second number: 92
Enter third number: 95
95 is bigger than 91 and 92
```

Conclusion:

The Conclusion by using nested if else loop to check which number is the biggest among three numbers.

Name of Student: __Sparsh Sharma____

Roll Number: _____**37**_______

Experiment No: **6**

Title:

Write a program that determines the grade of a student based on their marks of 5 subjects using if-else-if ladder.

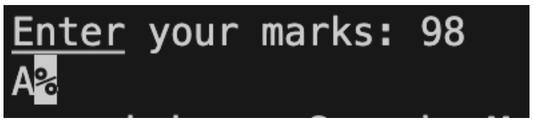
Theory:

Using if else-if ladder loop to check the marks given by user and printing the appropriate grade based on the marks. If marks>=90, grade-A. Marks>=80, grade-B. Marks>=70, grade-C. Marks>=60, grade-D. Else grade-F.

Code:

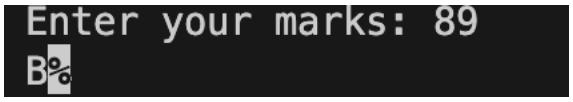
```
//Implement a program that determines the grade of a student based on their marks.

#include <iostream>
using namespace std;

int main() {
    float marks;
    cout << "Enter your marks: ";
    cin >> marks;
    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)

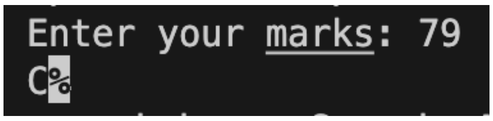


```
Enter your marks: 98
A%
```

Test Case: Any two (screenshot)



```
Enter your marks: 89
B%
```



```
Enter your marks: 79
C%
```

Conclusion:

The Conclusion is by using laddered if else-if loop, printing the grade based on the marks given by the user.

Name of Student: __Sparsh Sharma____

Roll Number: _____37_______

Experiment No: 7

Title:

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

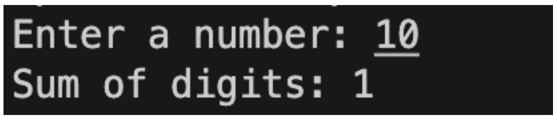
Theory:

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:

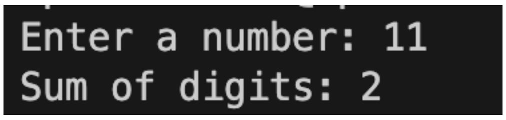
```
//program to find the sum of digits of a number until it becomes a 
single-digit number
#include<iostream>
using namespace std;
int main()
{
    int n,r;
    cout<<"Enter a number: ";
    cin>>n;
    while (n>=10)
    {
        int sum=0;
        while(n>0)
        {
            r=n%10;
            sum+=r;
            n/=10;
        }
        n=sum;
    }
    cout<<"Sum of digits: "<<n<<endl;
    return 0;
}
```

Output: (screenshot)

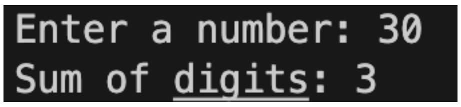


```
Enter a number: 10
Sum of digits: 1
```

Test Case: Any two (screenshot)



```
Enter a number: 11
Sum of digits: 2
```



```
Enter a number: 30
Sum of digits: 3
```

Conclusion:

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

Name of Student: __Sparsh Sharma____

Roll Number: ______37________

Experiment No: **8**

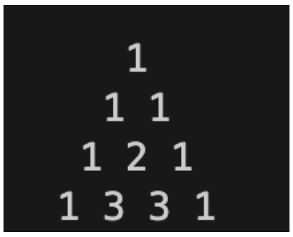
Title: Write a program to print a Pascal's triangle using nested loops.

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:

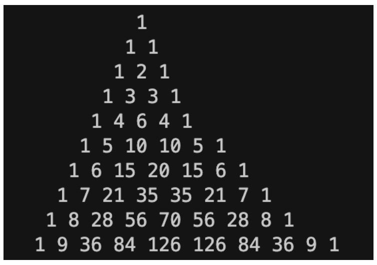
```
// pascal triangle
#include <iostream>
using namespace std;
int main()
{
    int n;
    cout << "Enter number: ";
    cin >> n;

    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):-

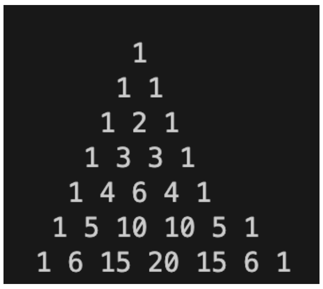


```
      1
     1 1
    1 2 1
   1 3 3 1
```

Test Case: Any two (screenshot)



```
          1
         1 1
        1 2 1
       1 3 3 1
      1 4 6 4 1
     1 5 10 10 5 1
    1 6 15 20 15 6 1
   1 7 21 35 35 21 7 1
  1 8 28 56 70 56 28 8 1
 1 9 36 84 126 126 84 36 9 1
```



```
       1
      1 1
     1 2 1
    1 3 3 1
   1 4 6 4 1
  1 5 10 10 5 1
 1 6 15 20 15 6 1
```

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.

Name of Student: __Sparsh Sharma____

Roll Number: ______37_______

Experiment No: 9

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

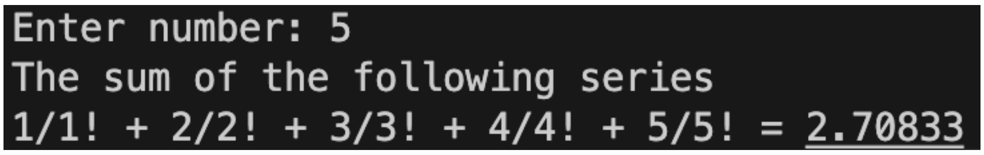
Theory: Using factorial function to calculate factorial of a number. Taking range from the user and using a for loop to calculate sum of the series using the factorial function to calculate the denominator and printing the sum at the end.

Code:

```
// sum of series 1/1! + 2/2! + 3/3! + ... + N/N! using nested
loops.
#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) {
```

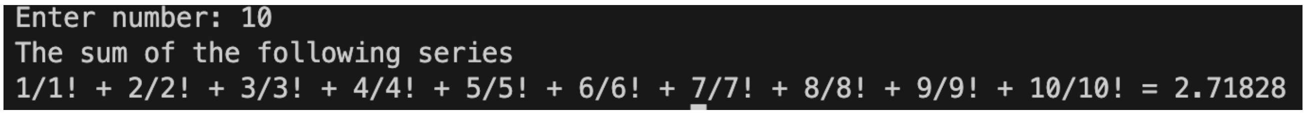
```cpp
            cout << i << "/" << i << "! + ";
        } else {
            cout << i << "/" << i << "! = ";
        }

        sum += double(i) / nFact;
    }

    cout << sum << endl;

    return 0;
}
```

Output: (screenshot)

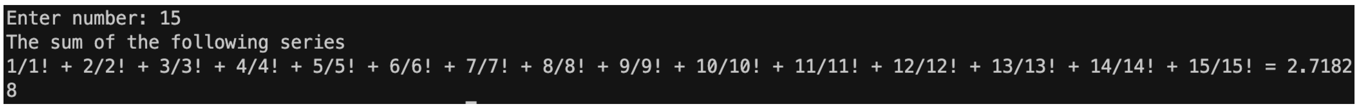


```
Enter number: 5
The sum of the following series
1/1! + 2/2! + 3/3! + 4/4! + 5/5! = 2.70833
```

Test Case: Any two (screenshot)



```
Enter number: 10
The sum of the following series
1/1! + 2/2! + 3/3! + 4/4! + 5/5! + 6/6! + 7/7! + 8/8! + 9/9! + 10/10! = 2.71828
```



```
Enter number: 15
The sum of the following series
1/1! + 2/2! + 3/3! + 4/4! + 5/5! + 6/6! + 7/7! + 8/8! + 9/9! + 10/10! + 11/11! + 12/12! + 13/13! + 14/14! + 15/15! = 2.7182
8
```

Conclusion: The by using nested for loops and recursive factorial function to calculate sum of the series till the range given by the user and printing the sum.

Name of Student: __Sparsh Sharma____

Roll Number: _____37_______

Experiment No: 10

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

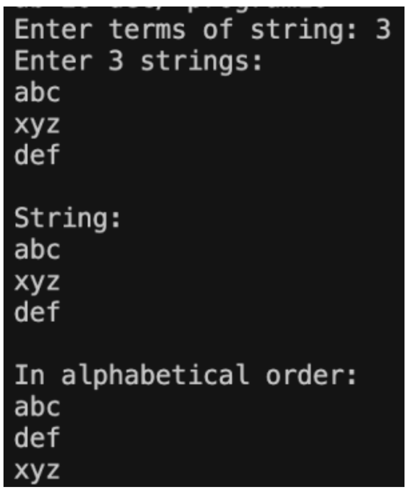
Theory:

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:

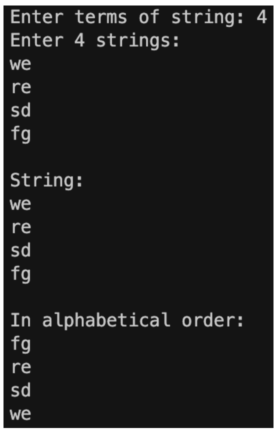
```
//program to create an array of strings and display them in 
alphabetical order
#include<iostream>
#include<string>
using namespace std;
int main()
{
    int n;
    cout<<"Enter terms of string: ";
    cin>>n;
    string str[n];
    cout<<"Enter "<<n<<" strings: "<<endl;
    for (int i=0; i<n; i++)
    {
        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)

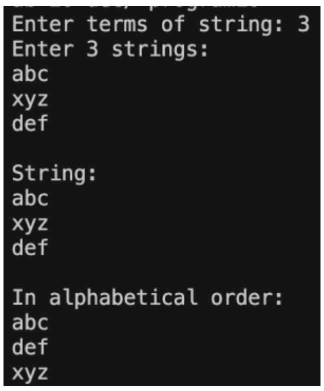


```
Enter terms of string: 3
Enter 3 strings:
abc
xyz
def

String:
abc
xyz
def

In alphabetical order:
abc
def
xyz
```

Test Case: Any two (screenshot)



```
Enter terms of string: 4
Enter 4 strings:
we
re
sd
fg

String:
we
re
sd
fg

In alphabetical order:
fg
re
sd
we
```



```
Enter terms of string: 3
Enter 3 strings:
abc
xyz
def

String:
abc
xyz
def

In alphabetical order:
abc
def
xyz
```

Conclusion:

The Conclusion is by using nested for loops and sorting the strings in the array in alphabetical order and printing the array.

Name of Student: __Sparsh Sharma____

Roll Number: _____37_______

Experiment No: 11

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

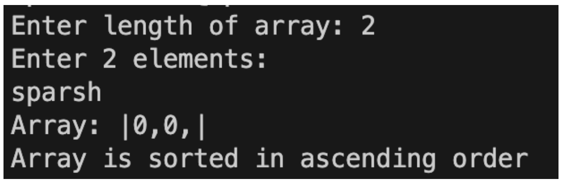
Theory:

Creating an array of integers of size given by the user and adding the numbers from the 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:

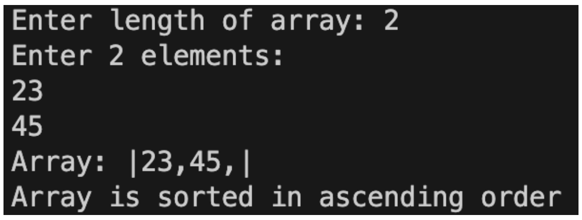
```cpp
//program to check if an array is sorted in ascending order
#include<iostream>
using namespace std;
int main()
{
    int n;bool sorted = true;
    cout<<"Enter length of array: ";
    cin>>n;
    int arr[n];
    cout<<"Enter "<<n<<" elements: "<<endl;
    for (int i=0; i<n; i++)
    {
        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)

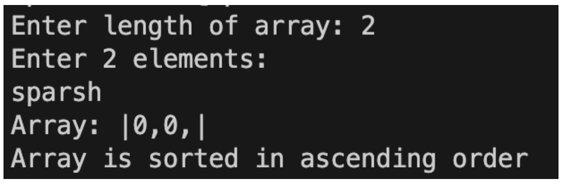


```
Enter length of array: 2
Enter 2 elements:
sparsh
Array: |0,0,|
Array is sorted in ascending order
```

Test Case: Any two (screenshot)



```
Enter length of array: 2
Enter 2 elements:
23
45
Array: |23,45,|
Array is sorted in ascending order
```



```
Enter length of array: 2
Enter 2 elements:
sparsh
Array: |0,0,|
Array is sorted in ascending order
```

Conclusion:

The Conclusion is 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.

Name of Student: __Sparsh Sharma____

Roll Number: _____37_______

Experiment No: 12

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

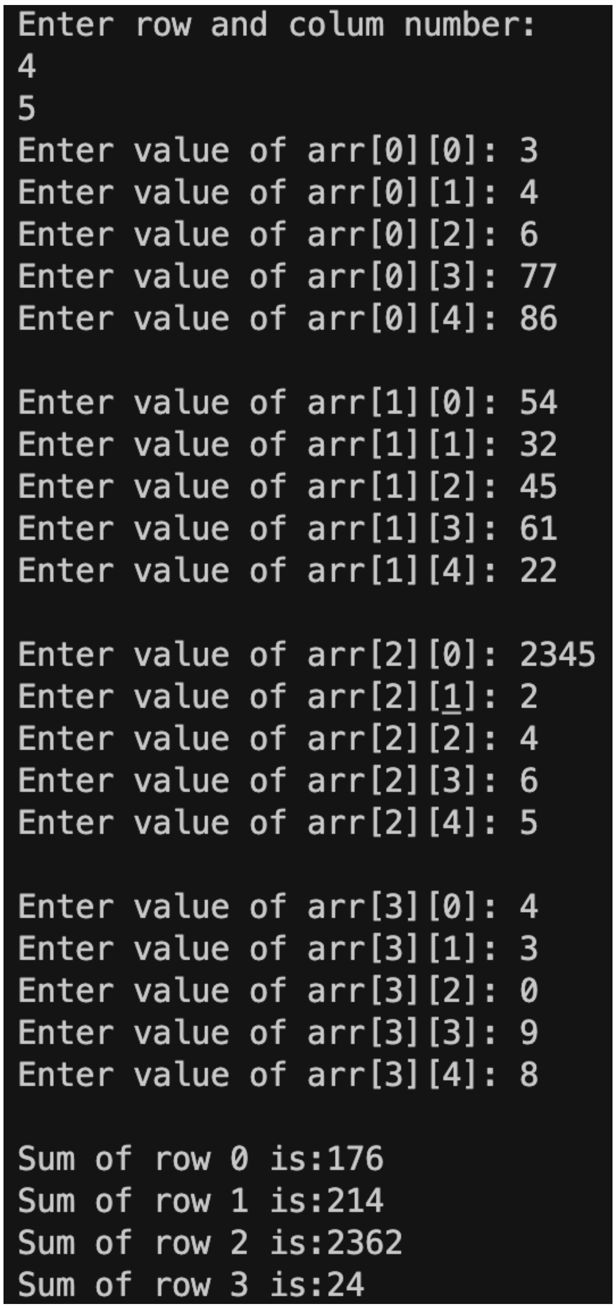
Theory:

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:

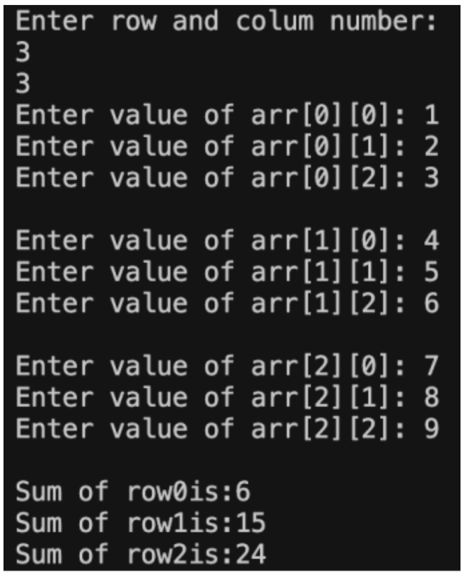
```
//Calculate the sum of elements in each row of a matrix.
#include <iostream>
using namespace std;

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)

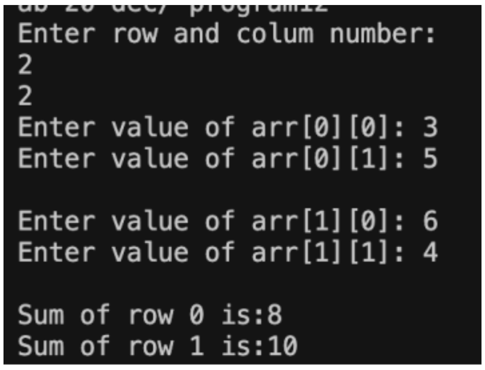


```
Enter row and colum number:
4
5
Enter value of arr[0][0]: 3
Enter value of arr[0][1]: 4
Enter value of arr[0][2]: 6
Enter value of arr[0][3]: 77
Enter value of arr[0][4]: 86

Enter value of arr[1][0]: 54
Enter value of arr[1][1]: 32
Enter value of arr[1][2]: 45
Enter value of arr[1][3]: 61
Enter value of arr[1][4]: 22

Enter value of arr[2][0]: 2345
Enter value of arr[2][1]: 2
Enter value of arr[2][2]: 4
Enter value of arr[2][3]: 6
Enter value of arr[2][4]: 5

Enter value of arr[3][0]: 4
Enter value of arr[3][1]: 3
Enter value of arr[3][2]: 0
Enter value of arr[3][3]: 9
Enter value of arr[3][4]: 8

Sum of row 0 is:176
Sum of row 1 is:214
Sum of row 2 is:2362
Sum of row 3 is:24
```

Test Case: Any two (screenshot)



```
Enter row and colum number:
3
3
Enter value of arr[0][0]: 1
Enter value of arr[0][1]: 2
Enter value of arr[0][2]: 3

Enter value of arr[1][0]: 4
Enter value of arr[1][1]: 5
Enter value of arr[1][2]: 6

Enter value of arr[2][0]: 7
Enter value of arr[2][1]: 8
Enter value of arr[2][2]: 9

Sum of row0is:6
Sum of row1is:15
Sum of row2is:24
```



```
Enter row and colum number:
2
2
Enter value of arr[0][0]: 3
Enter value of arr[0][1]: 5

Enter value of arr[1][0]: 6
Enter value of arr[1][1]: 4

Sum of row 0 is:8
Sum of row 1 is:10
```

Conclusion:

The Conclusion is 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.

Name of Student: __Sparsh Sharma____

Roll Number: _____37_______

Experiment No: 13

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

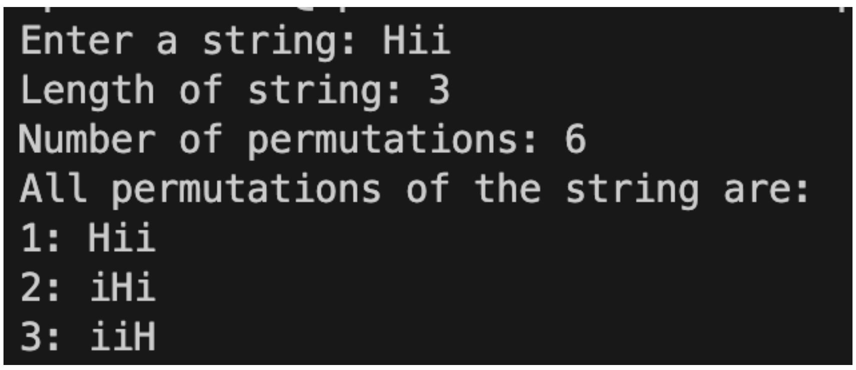
Theory:

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:

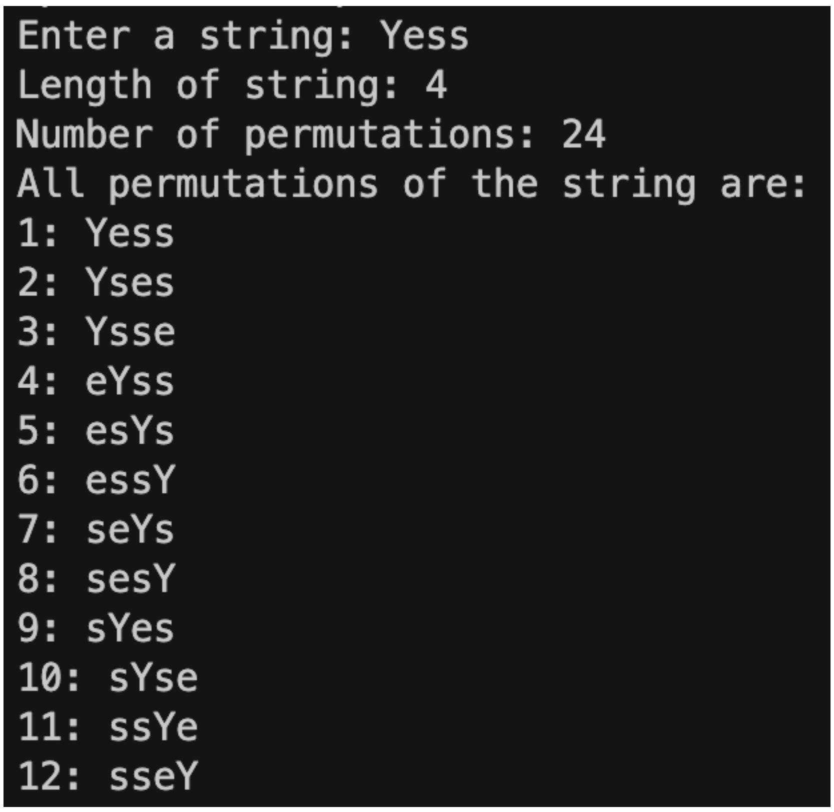
```
//program to generate all possible permutations of a string
#include <iostream>
#include <string>
using namespace std;

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;
        }
        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)



```
Enter a string: Hii
Length of string: 3
Number of permutations: 6
All permutations of the string are:
1: Hii
2: iHi
3: iiH
```

Test Case: Any two (screenshot)



```
Enter a string: Yess
Length of string: 4
Number of permutations: 24
All permutations of the string are:
1: Yess
2: Yses
3: Ysse
4: eYss
5: esYs
6: essY
7: seYs
8: sesY
9: sYes
10: sYse
11: ssYe
12: sseY
```

Conclusion: The Conclusion is by using factorial and permutation function to calculate all possible permutations of a string given by the user.

Name of Student: __Sparsh Sharma____

Roll Number: ______37_______

Experiment No: 14

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

```
*****

* *

* *

* *

*****
```

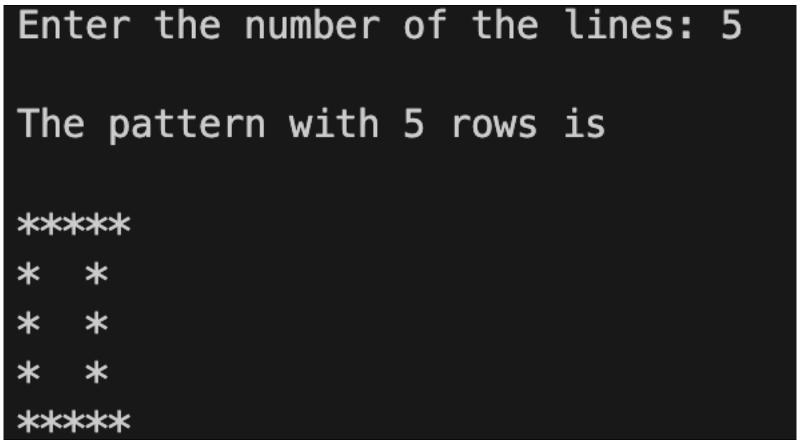
Theory:

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

Code:

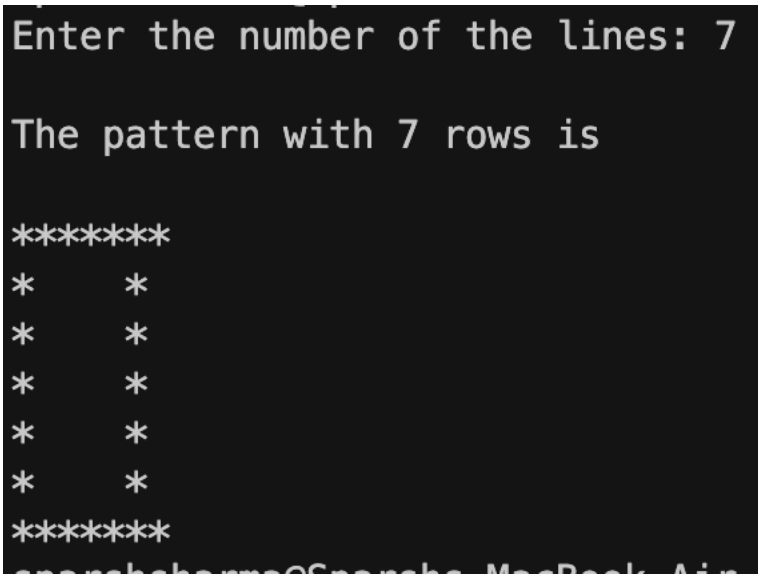
```cpp
// rectangular star pattern
#include <iostream>
using namespace std;

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)



```
Enter the number of the lines: 5

The pattern with 5 rows is

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

Test Case: Any two (screenshot)



```
Enter the number of the lines: 7

The pattern with 7 rows is

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

Conclusion:

The Conclusion is by using nested for loops and if else statements to print stars in specific rows and columns to print a pattern.

Name of Student: __Sparsh Sharma____

Roll Number: _____37_______

Experiment No: 15

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

1

232

34543

4567654

34543

232

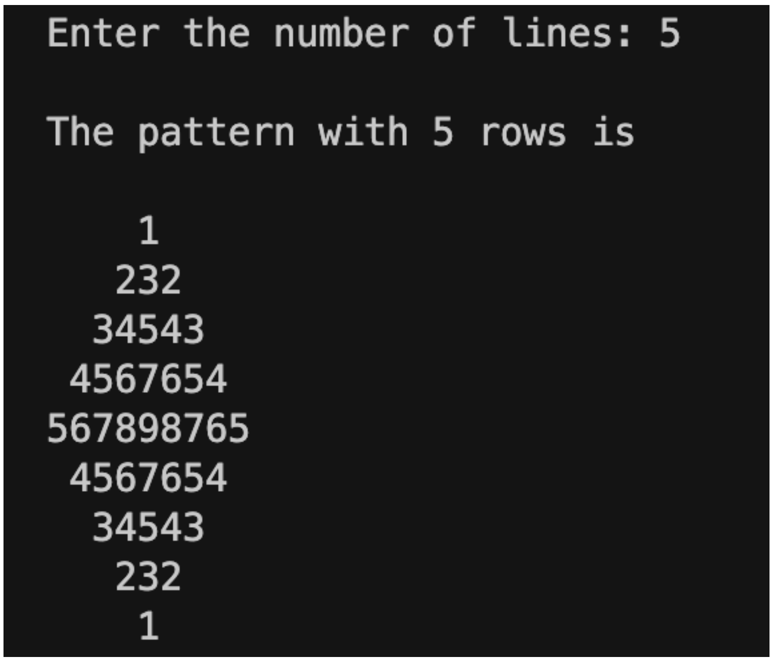
Theory:

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:

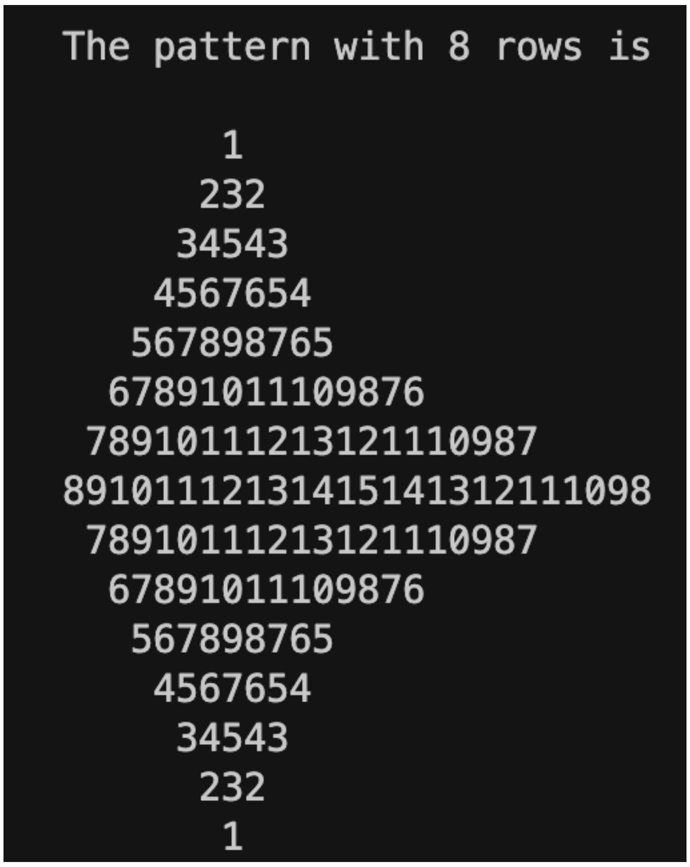
```
// pyramid number pattern
#include <iostream>
using namespace std;


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)



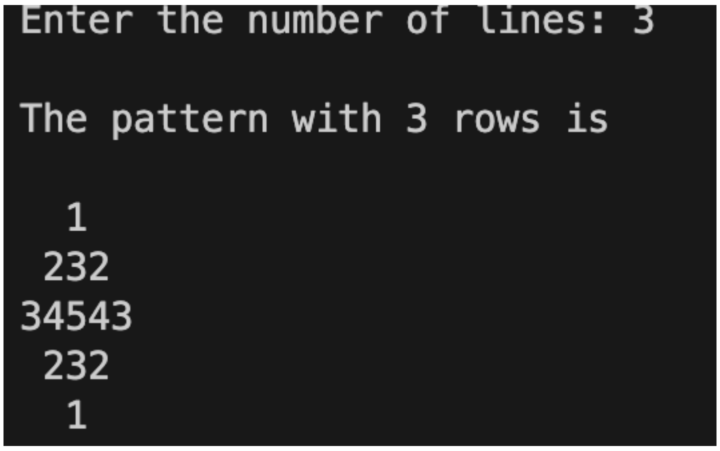
```
Enter the number of lines: 5

The pattern with 5 rows is

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

Test Case: Any two (screenshot)



```
The pattern with 8 rows is

      1
     232
    34543
   4567654
  567898765
 67891011109876
 78910111213121110987
8910111213141514131211109 8
```

```
The pattern with 8 rows is

       1
      232
     34543
    4567654
   567898765
  67891011109876
 78910111213121110987
891011121314151413121110 98
```



```
Enter the number of lines: 3

The pattern with 3 rows is

  1
 232
34543
 232
  1
```

Conclusion:

The Conclusion is 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.

Name of Student: __Sparsh Sharma____

Roll Number: _____**37**_______

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 parameterised constructor for the Product class to initialise the attributes when a new product is added to the inventory.

Theory:

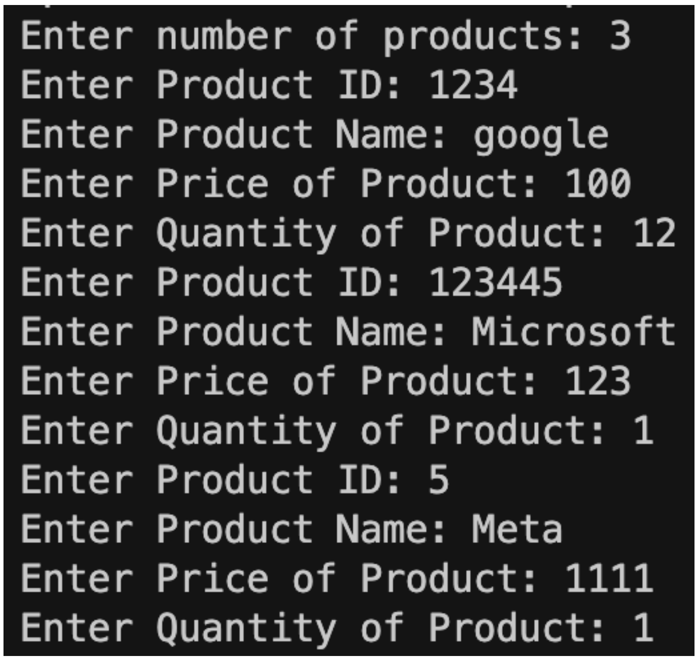
A class is a blueprint for objects. It consists of attributes and methods. An object is an instance of a class. It has a copy of the attributes and shares the methods with other objects. A parameterised constructor is used to initialise the attributes when an object is created with some arguments.

Code:

```
// store inventory management system
#include <iostream>
#include <string>
using namespace std;

class Product
{
```

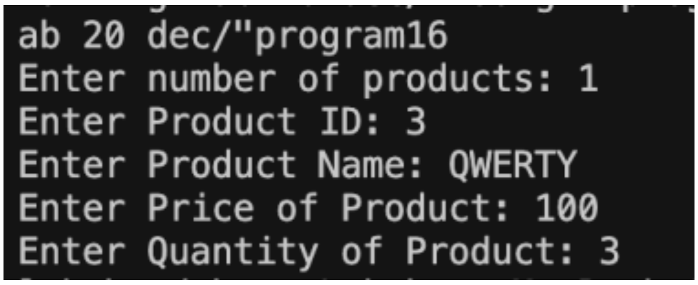
```
private:
    int prod_id;
    string prod_name;
    float price;
    int quantity;

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

Output: (screenshot)

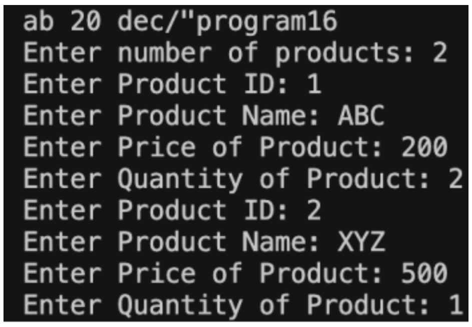


```
Enter number of products: 3
Enter Product ID: 1234
Enter Product Name: google
Enter Price of Product: 100
Enter Quantity of Product: 12
Enter Product ID: 123445
Enter Product Name: Microsoft
Enter Price of Product: 123
Enter Quantity of Product: 1
Enter Product ID: 5
Enter Product Name: Meta
Enter Price of Product: 1111
Enter Quantity of Product: 1
```

Test Case: Any two (screenshot)



```
ab 20 dec/"program16
Enter number of products: 1
Enter Product ID: 3
Enter Product Name: QWERTY
Enter Price of Product: 100
Enter Quantity of Product: 3
```



```
ab 20 dec/"program16
Enter number of products: 2
Enter Product ID: 1
Enter Product Name: ABC
Enter Price of Product: 200
Enter Quantity of Product: 2
Enter Product ID: 2
Enter Product Name: XYZ
Enter Price of Product: 500
Enter Quantity of Product: 1
```

Conclusion:

The Conclusion is by using classes and list of objects of size given by the user and using a parameterised constructor to initialise attributes when an object is created with some arguments given by the user.

Name of Student: __Sparsh Sharma____

Roll Number: _____**37**_______

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:

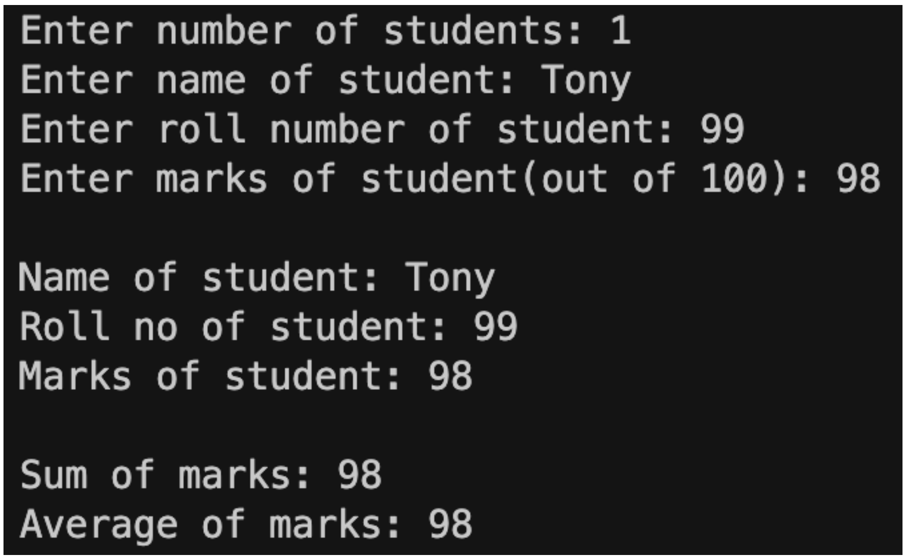
Using parameterised constructor to show details of students. Using for loop to take student details from the user and calculating average of students.

Code:

```
// program to manage student records
#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()
    {
```

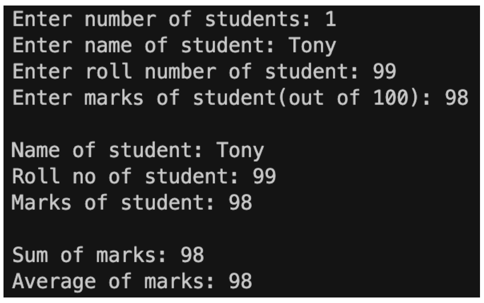
```cpp
        cout << endl
             << "Name of student: " << name << endl;
        cout << "Roll no of student: " << roll << endl;
        cout << "Marks of student: " << marks << endl;
    }
};

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)

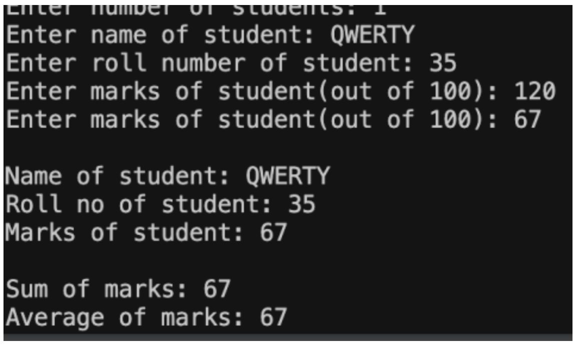


```
Enter number of students: 1
Enter name of student: Tony
Enter roll number of student: 99
Enter marks of student(out of 100): 98

Name of student: Tony
Roll no of student: 99
Marks of student: 98

Sum of marks: 98
Average of marks: 98
```

Test Case: Any two (screenshot)



```
Enter number of students: 1
Enter name of student: Tony
Enter roll number of student: 99
Enter marks of student(out of 100): 98

Name of student: Tony
Roll no of student: 99
Marks of student: 98

Sum of marks: 98
Average of marks: 98
```



```
Enter number of students: 1
Enter name of student: QWERTY
Enter roll number of student: 35
Enter marks of student(out of 100): 120
Enter marks of student(out of 100): 67

Name of student: QWERTY
Roll no of student: 35
Marks of student: 67

Sum of marks: 67
Average of marks: 67
```

Conclusion:

The Conclusion is 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.

Name of Student: __Sparsh Sharma____

Roll Number: _____37_______

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:

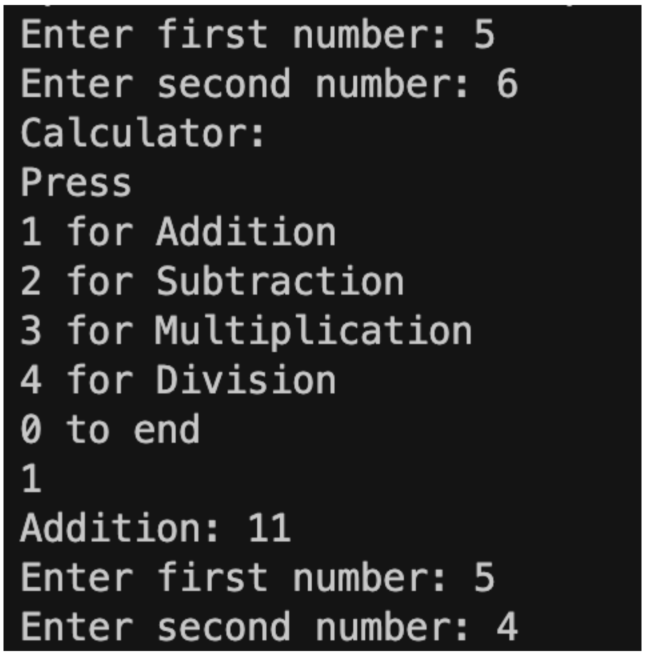
Creating a class and making a method for getting values from the user and printing a menu of different operations to perform and performing the user given operation on the values and printing the result using while loop.

Code:

```cpp
// program to simulate a simple calculator
#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;
    }
    float division(float x, float y)
    {
        if (x == 0 || y == 0)
        {
            cout << "Invalid number" << endl;
            return 0;
        }
        else
        {
            cout << "Division: " << x / y << endl;
```

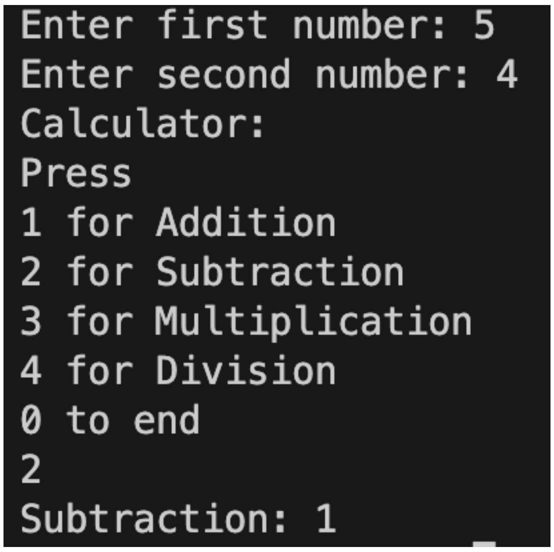
```
                return 0;
            }
        }
};

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

Output: (screenshot)

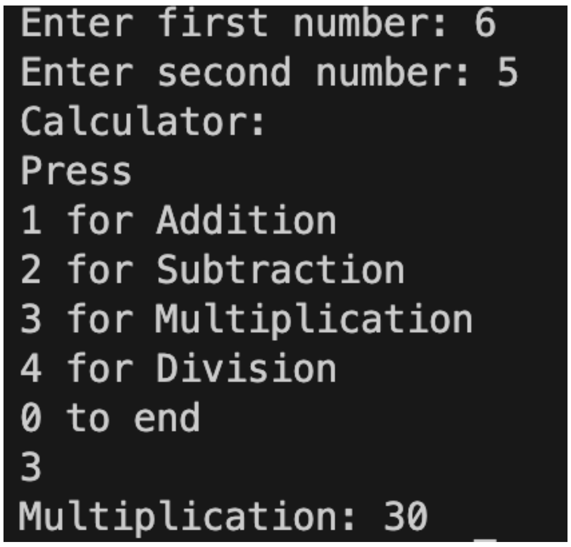


```
Enter first number: 5
Enter second number: 6
Calculator:
Press
1 for Addition
2 for Subtraction
3 for Multiplication
4 for Division
0 to end
1
Addition: 11
Enter first number: 5
Enter second number: 4
```

Test Case: Any two (screenshot)



```
Enter first number: 5
Enter second number: 4
Calculator:
Press
1 for Addition
2 for Subtraction
3 for Multiplication
4 for Division
0 to end
2
Subtraction: 1
```



```
Enter first number: 6
Enter second number: 5
Calculator:
Press
1 for Addition
2 for Subtraction
3 for Multiplication
4 for Division
0 to end
3
Multiplication: 30
```

Conclusion:

The Conclusion Is 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.

Name of Student: __Sparsh Sharma____

Roll Number: _____37_______

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

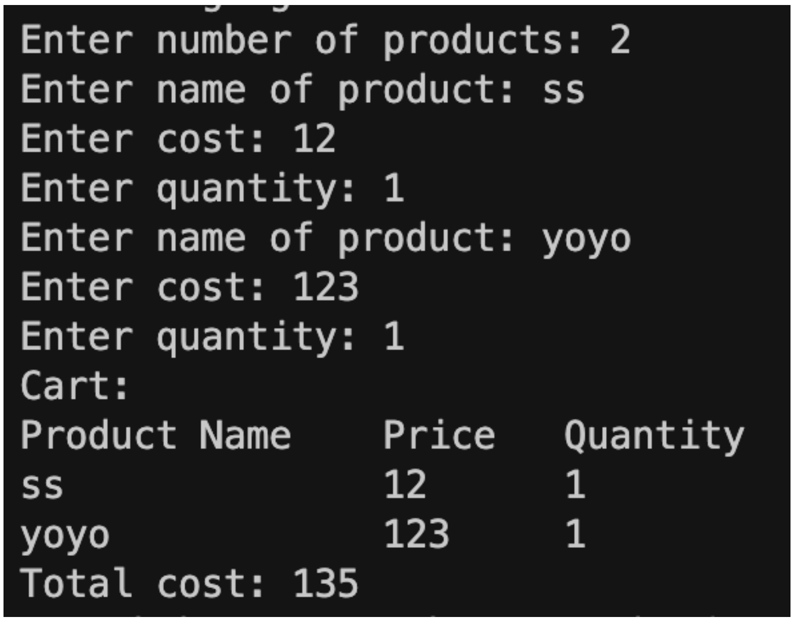
the contents of the cart.

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:

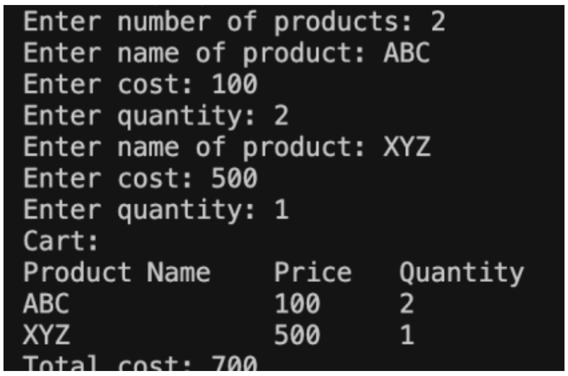
```
// online shop simulator
#include <iostream>
using namespace std;

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)

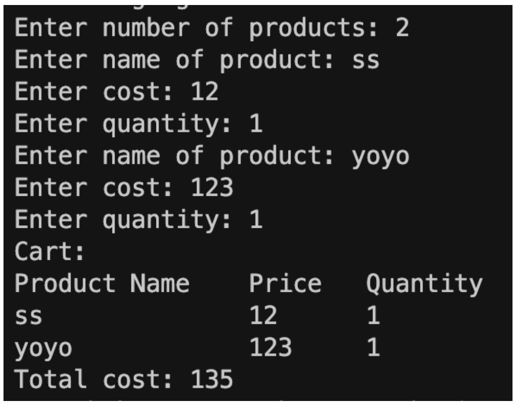


```
Enter number of products: 2
Enter name of product: ss
Enter cost: 12
Enter quantity: 1
Enter name of product: yoyo
Enter cost: 123
Enter quantity: 1
Cart:
Product Name    Price   Quantity
ss              12      1
yoyo            123     1
Total cost: 135
```

Test Case: Any two (screenshot)



```
Enter number of products: 2
Enter name of product: ABC
Enter cost: 100
Enter quantity: 2
Enter name of product: XYZ
Enter cost: 500
Enter quantity: 1
Cart:
Product Name    Price   Quantity
ABC             100     2
XYZ             500     1
Total cost: 700
```



```
Enter number of products: 2
Enter name of product: ss
Enter cost: 12
Enter quantity: 1
Enter name of product: yoyo
Enter cost: 123
Enter quantity: 1
Cart:
Product Name    Price   Quantity
ss              12      1
yoyo            123     1
Total cost: 135
```

Conclusion:

The Conclusion is 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.

Name of Student: __Sparsh Sharma____

Roll Number: _____37_______

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:

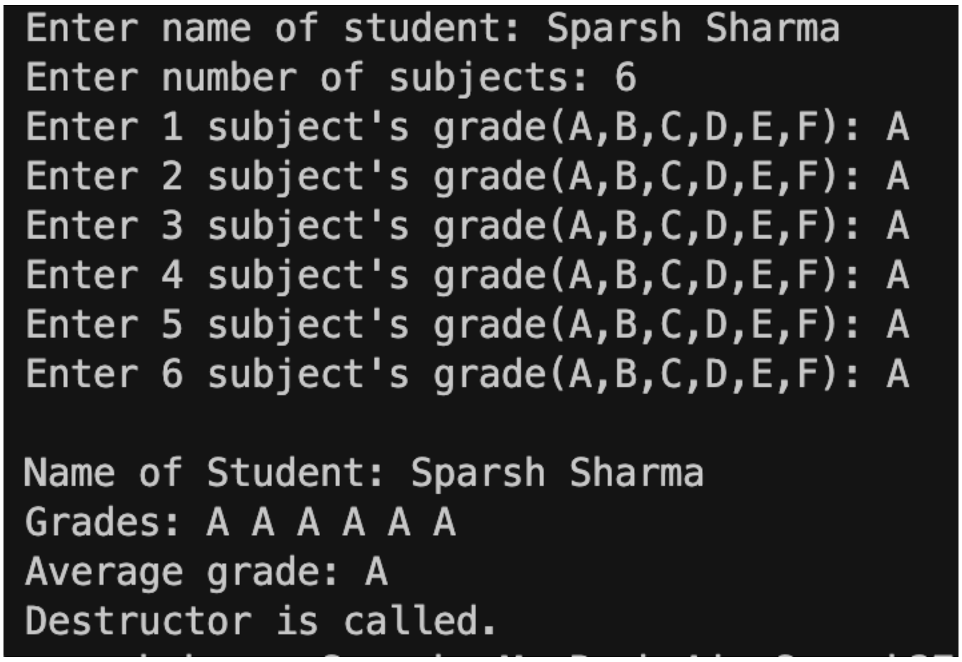
Using constructor to get name of student and methods to get grades of the student and store in an array and using for loop to calculate sum of grades and display average grade and student name.

Code:// program to manage grades of students in a classroom

```cpp
#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)
```

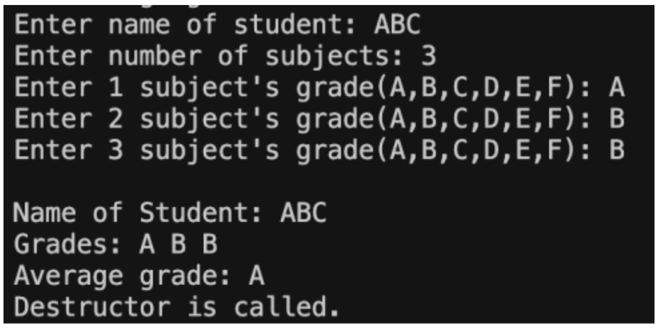
```cpp
        {
            grades = 'C';
        }
        else if (avg > 60 && avg <= 70)
        {
            grades = 'D';
        }
        else if (avg > 50 && avg <= 60)
        {
            grades = 'E';
        }
        else
        {
            grades = 'F';
        }
    }
    void showDetails()
    {
        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)

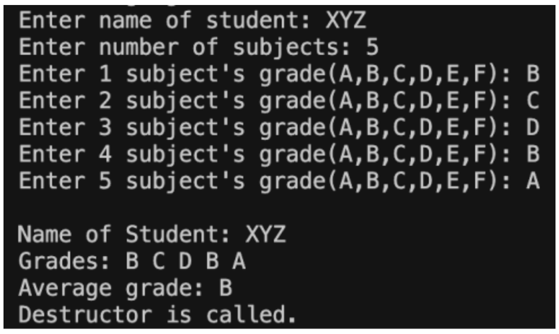


```
Enter name of student: Sparsh Sharma
Enter number of subjects: 6
Enter 1 subject's grade(A,B,C,D,E,F): A
Enter 2 subject's grade(A,B,C,D,E,F): A
Enter 3 subject's grade(A,B,C,D,E,F): A
Enter 4 subject's grade(A,B,C,D,E,F): A
Enter 5 subject's grade(A,B,C,D,E,F): A
Enter 6 subject's grade(A,B,C,D,E,F): A

Name of Student: Sparsh Sharma
Grades: A A A A A A
Average grade: A
Destructor is called.
```

Test Case: Any two (screenshot)



```
Enter name of student: ABC
Enter number of subjects: 3
Enter 1 subject's grade(A,B,C,D,E,F): A
Enter 2 subject's grade(A,B,C,D,E,F): B
Enter 3 subject's grade(A,B,C,D,E,F): B

Name of Student: ABC
Grades: A B B
Average grade: A
Destructor is called.
```



```
Enter name of student: XYZ
Enter number of subjects: 5
Enter 1 subject's grade(A,B,C,D,E,F): B
Enter 2 subject's grade(A,B,C,D,E,F): C
Enter 3 subject's grade(A,B,C,D,E,F): D
Enter 4 subject's grade(A,B,C,D,E,F): B
Enter 5 subject's grade(A,B,C,D,E,F): A

Name of Student: XYZ
Grades: B C D B A
Average grade: B
Destructor is called.
```

Conclusion: The Conclusion is 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