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

# C++ PROGRAMMING LAB

## Prepared by:

Name of Student:Aayush Ajit Chounkar

Roll No: 04

Batch: 2023-27

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

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

| | |
|---|---|
| | 34543<br>4567654<br>34543<br>232 |
| 6 | Write a program to creating an inventory management system for a small store. The system should use object-oriented principles in C++. Your program should have the following features:<br>• Create a **Product** class that represents a product in the inventory. Each **Product** object should have the following attributes:<br>• Product ID (an integer)<br>• Product Name (a string)<br>• Price (a floating-point number)<br>• Quantity in stock (an integer)<br>• Implement a parameterized constructor for the **Product** class to initialize the attributes when a new product is added to the inventory. |
| 7 | Write a program to manage student records. Create a class Student with attributes such as name, roll number, and marks. Implement methods for displaying student details, adding new students, and calculating the average marks of all students in the record system. |
| 8 | Write a program that implements a basic calculator. Use a class Calculator with methods to perform addition, subtraction, multiplication, and division of two numbers. The program should allow the user to input two numbers and select an operation to perform. |
| 9 | Write a program to simulate a simple online shop. Create a class Product with attributes like name, price, and quantity in stock. Implement methods for adding products to the shopping cart, calculating the total cost, and displaying the contents of the cart. |
| 0 | Write a program to manage student grades for a classroom. Create a class Student with attributes for student name and an array to store grades. Implement methods for adding grades, calculating the average grade, and displaying the student's name and grades. Use constructors and destructors to initialize and release resources. |
| | |

**Name of Student:Aayush  Chounkar**

**Roll Number: 04**

**Experiment No:01**

---

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

**Theory:**

- **Quadratic equations are of the form ax^2 + bx + c = 0, with roots given by (-b ± √(b^2 - 4ac)) / (2a).**
- **The discriminant (b^2 - 4ac) determines the nature of roots: positive for real roots, zero for repeated roots, and negative for imaginary roots**

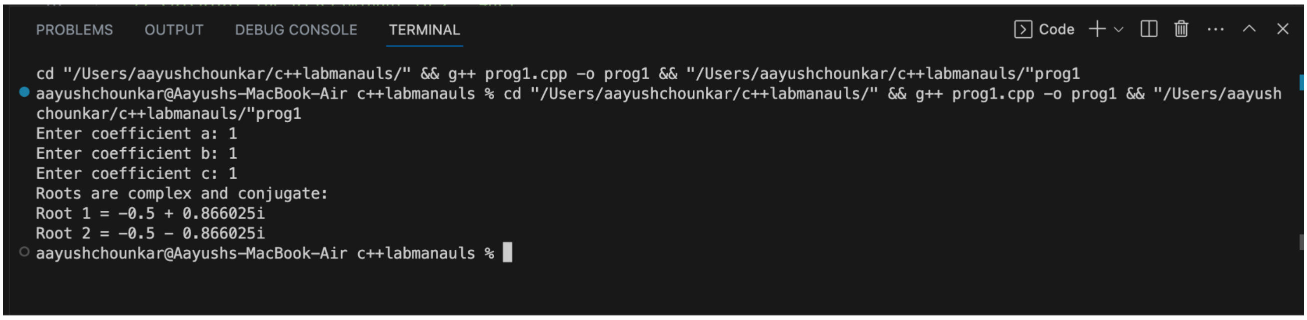
**Code:**

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

int main() {
   double a, b, c;

   // Input coefficients from the user
   cout << "Enter coefficient a: ";
   cin >> a;

   cout << "Enter coefficient b: ";
   cin >> b;

   cout << "Enter coefficient c: ";
```

```cpp
cin >> c;

// Calculate the discriminant (b^2 - 4ac)
double discriminant = b * b - 4 * a * c;

// Check the discriminant to determine the nature of roots
if (discriminant > 0) {
    // Two distinct real roots
    double root1 = (-b + sqrt(discriminant)) / (2 * a);
    double root2 = (-b - sqrt(discriminant)) / (2 * a);

    cout << "Roots are real and distinct:" << endl;
    cout << "Root 1 = " << root1 << endl;
    cout << "Root 2 = " << root2 << endl;
} else if (discriminant == 0) {
    // One real root (double root)
    double root = -b / (2 * a);

    cout << "Roots are real and identical:" <<endl;
    cout << "Root = " << root << endl;
} else {
    // Complex roots
    double realPart = -b / (2 * a);
    double imaginaryPart = sqrt(-discriminant) / (2 * a);

    cout << "Roots are complex and conjugate:" << endl;
    cout << "Root 1 = " << realPart << " + " << imaginaryPart << "i" << endl;
    cout << "Root 2 = " << realPart << " - " << imaginaryPart << "i" << endl;
}
```

```
    return 0;

}
```

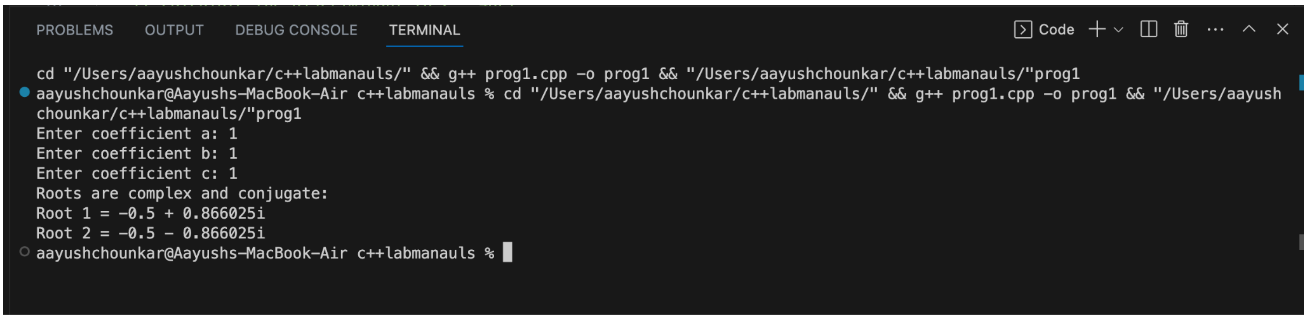
## Output: (screenshot)



```
cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog1.cpp -o prog1 && "/Users/aayushchounkar/c++labmanauls/"prog1
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog1.cpp -o prog1 && "/Users/aayush
chounkar/c++labmanauls/"prog1
Enter coefficient a: 1
Enter coefficient b: 1
Enter coefficient c: 1
Roots are complex and conjugate:
Root 1 = -0.5 + 0.866025i
Root 2 = -0.5 - 0.866025i
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % 
```

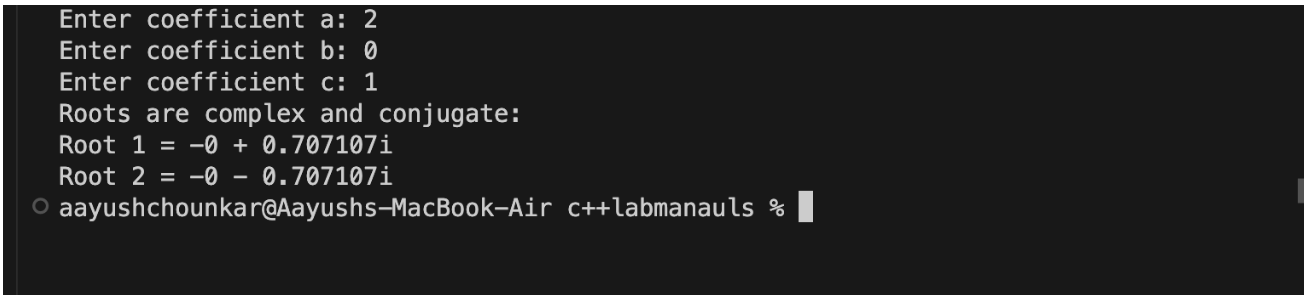
## Test Case: Any two (screenshot)

**1.**



```
cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog1.cpp -o prog1 && "/Users/aayushchounkar/c++labmanauls/"prog1
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog1.cpp -o prog1 && "/Users/aayush
chounkar/c++labmanauls/"prog1
Enter coefficient a: 1
Enter coefficient b: 1
Enter coefficient c: 1
Roots are complex and conjugate:
Root 1 = -0.5 + 0.866025i
Root 2 = -0.5 - 0.866025i
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % 
```

**2.**



```
Enter coefficient a: 2
Enter coefficient b: 0
Enter coefficient c: 1
Roots are complex and conjugate:
Root 1 = -0 + 0.707107i
Root 2 = -0 - 0.707107i
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % 
```

**Conclusion:**

- **The program is working accurately and The program accurately calculates the roots of a quadratic equation, providing real, repeated, or imaginary solutions based on the discriminant**

**Name of Student:Aayush  Chounkar**

**Roll Number: 04**

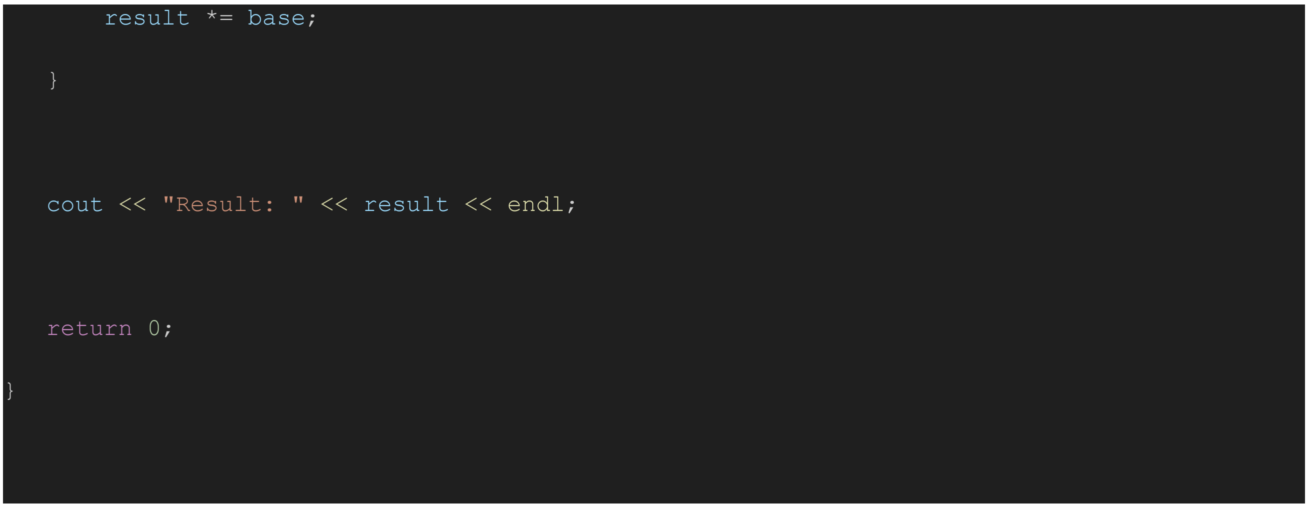
**Experiment No:02**

- **TITLE-**Write a program to calculate the power of a number using a loop.

- THEORY-
- Power calculation involves multiplying the base by itself for the specified exponent using a loop.
- The loop iterates through the exponent, updating the result by multiplying it with the base in each iteration

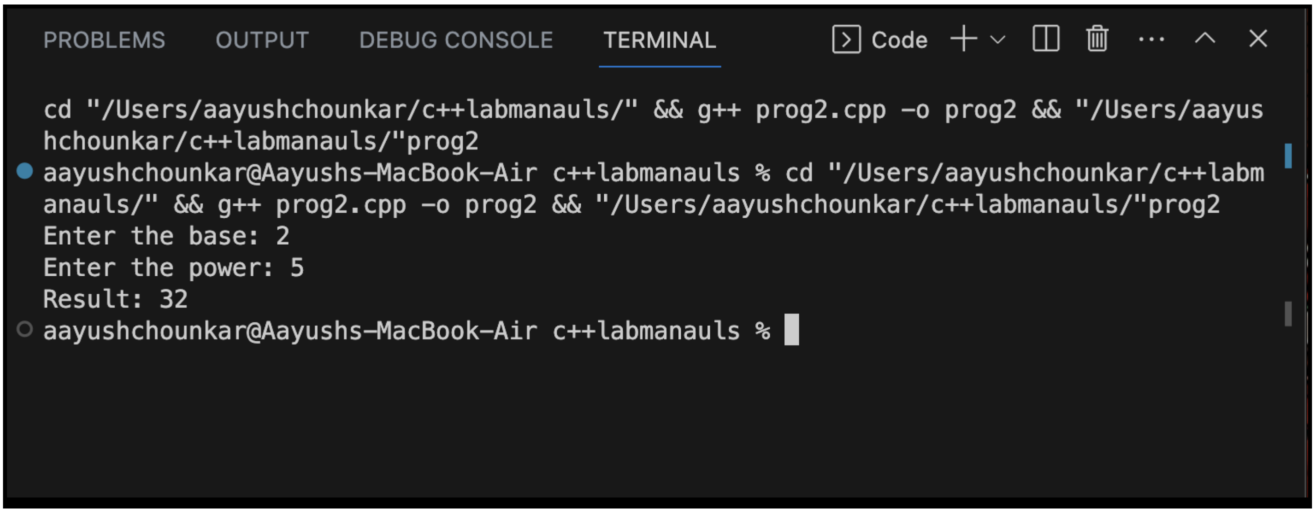
- CODE-

```
#include <iostream>


using namespace std;


int main() {

   int base, power, result = 1;


   cout << "Enter the base: ";

   cin >> base;


   cout << "Enter the power: ";

   cin >> power;


   // Calculate power using a loop

   for (int i = 0; i < power; ++i) {
```



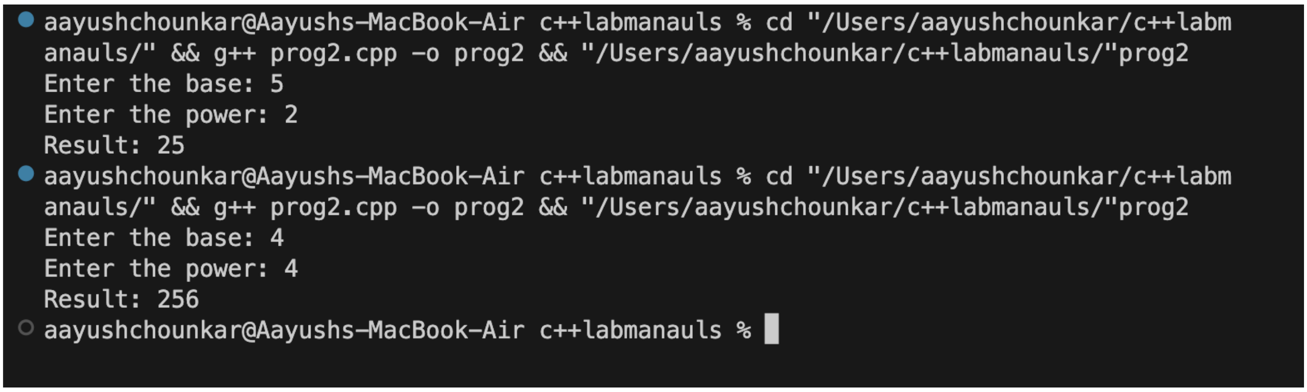
```
        result *= base;
    }

    cout << "Result: " << result << endl;

    return 0;
}
```

- OUTPUT-



```
cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog2.cpp -o prog2 && "/Users/aayus
hchounkar/c++labmanauls/"prog2
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labm
anauls/" && g++ prog2.cpp -o prog2 && "/Users/aayushchounkar/c++labmanauls/"prog2
Enter the base: 2
Enter the power: 5
Result: 32
aayushchounkar@Aayushs-MacBook-Air c++labmanauls %
```

- TEST CASE-



```
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labm
anauls/" && g++ prog2.cpp -o prog2 && "/Users/aayushchounkar/c++labmanauls/"prog2
Enter the base: 5
Enter the power: 2
Result: 25
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labm
anauls/" && g++ prog2.cpp -o prog2 && "/Users/aayushchounkar/c++labmanauls/"prog2
Enter the base: 4
Enter the power: 4
Result: 256
aayushchounkar@Aayushs-MacBook-Air c++labmanauls %
```

CONCLUSION-

The program effectively computes the power of a number using a loop, providing the result by iteratively multiplying the base with itself according to the given exponent.

**Name of Student:Aayush  Chounkar**

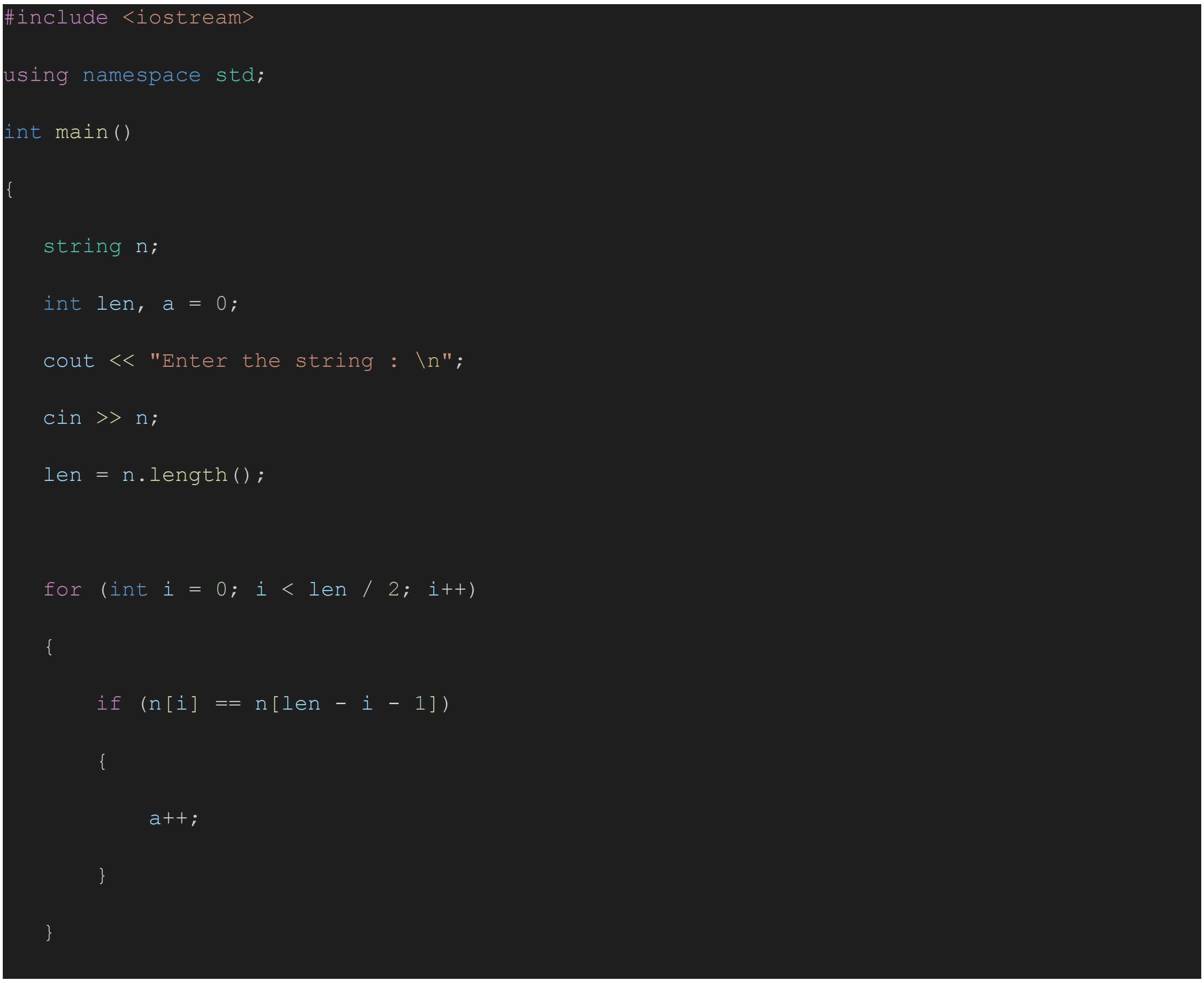
**Roll Number: 04**

**Experiment No:03**

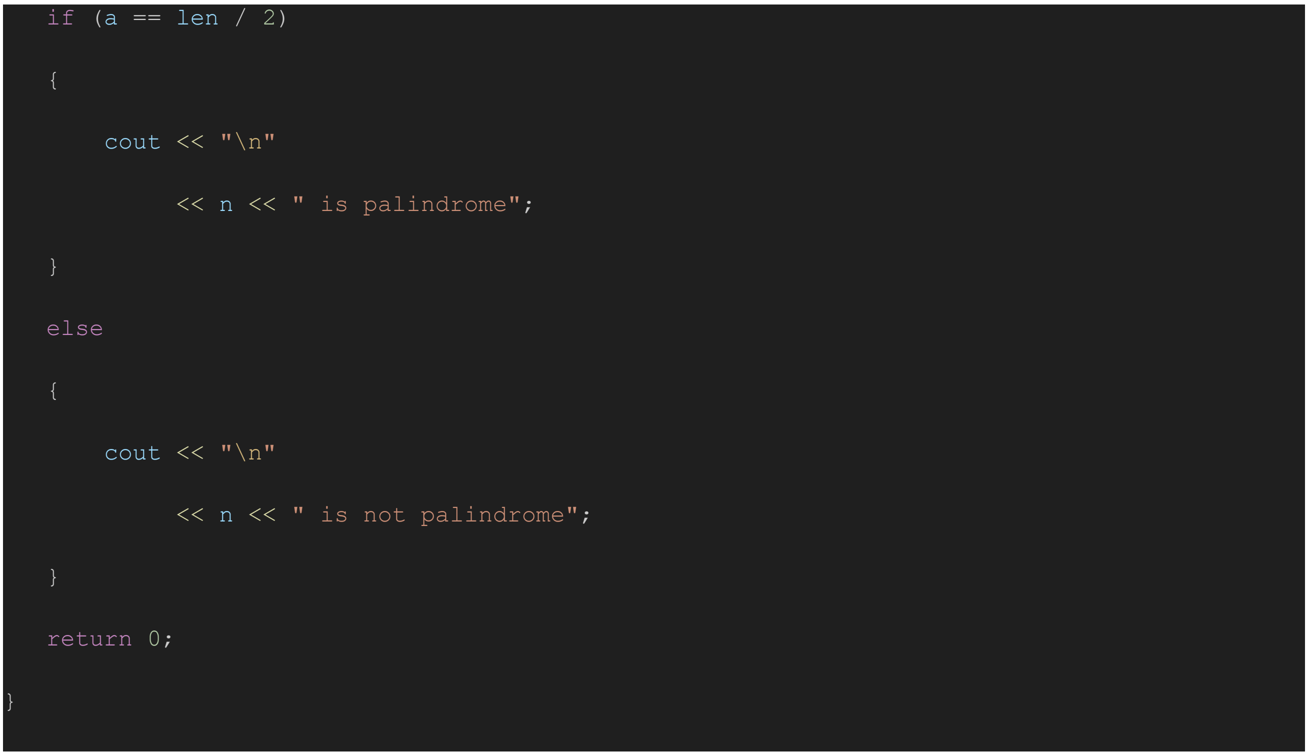
- TITLE-Write a program to check if a given string, is a palindrome.

- THEORY -
- • A palindrome is a string that reads the same backward as forward.
- •The program compares characters from both ends, moving towards the center, to determine if the given string is a palindrome.

- CODE-

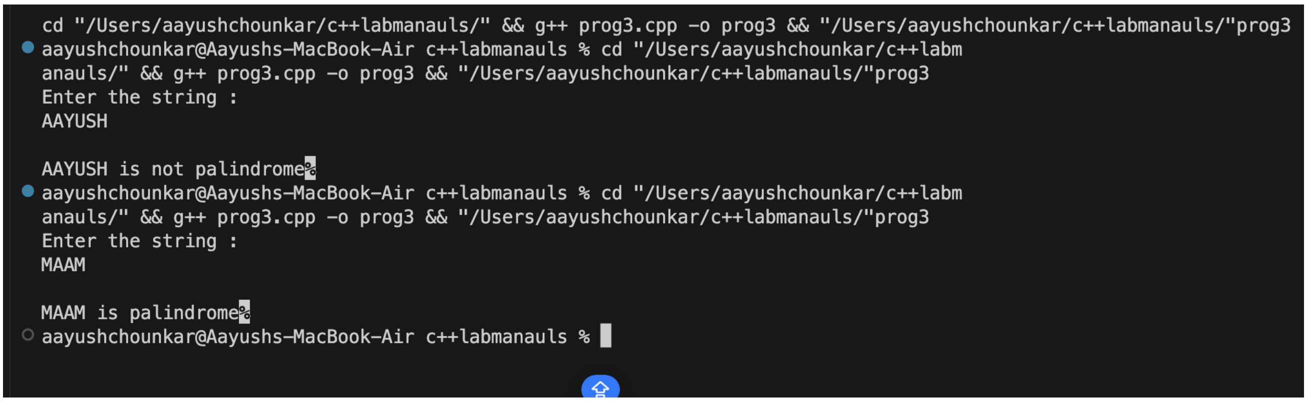


```
#include <iostream>

using namespace std;

int main()

{

   string n;

   int len, a = 0;

   cout << "Enter the string : \n";

   cin >> n;

   len = n.length();


   for (int i = 0; i < len / 2; i++)

   {

       if (n[i] == n[len - i - 1])

       {

           a++;

       }

   }
```



```
    if (a == len / 2)

    {

        cout << "\n"

             << n << " is palindrome";

    }

    else

    {

        cout << "\n"

             << n << " is not palindrome";

    }

    return 0;

}
```

- TEST CASE-



```
 cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog3.cpp -o prog3 && "/Users/aayushchounkar/c++labmanauls/"prog3
● aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labm
 anauls/" && g++ prog3.cpp -o prog3 && "/Users/aayushchounkar/c++labmanauls/"prog3
 Enter the string :
 AAYUSH

 AAYUSH is not palindrome%
● aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labm
 anauls/" && g++ prog3.cpp -o prog3 && "/Users/aayushchounkar/c++labmanauls/"prog3
 Enter the string :
 MAAM

 MAAM is palindrome%
○ aayushchounkar@Aayushs-MacBook-Air c++labmanauls % 
```

CONCLUSION-

The program is properly checking weather the string is palindrome or not .

By comparing characters from start to end .

**Name of Student:Aayush  Chounkar**

**Roll Number: 04**

**Experiment No:04**

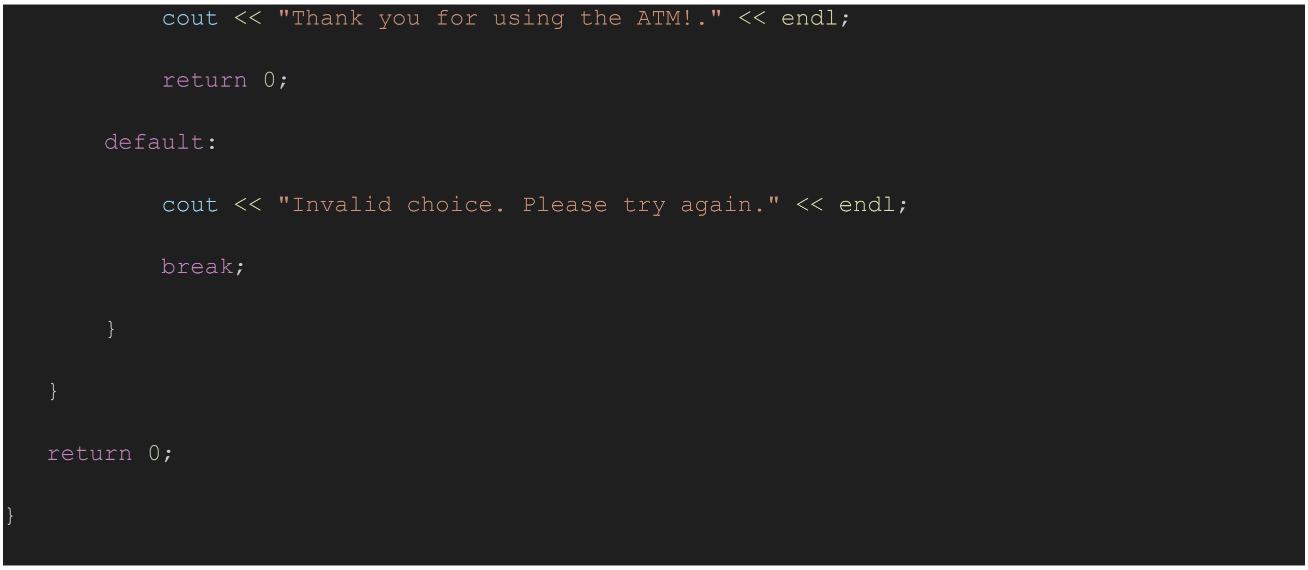
- TITLE- Write a program that simulates a simple ATM machine, allowing users to check their balance, deposit, or withdraw money using a switch statement.

- THEORY-
- The program uses a switch statement to offer options like checking balance, depositing, and withdrawing money in a simple ATM simulation.
- Each case in the switch statement executes the corresponding operation based on user input

- CODE-

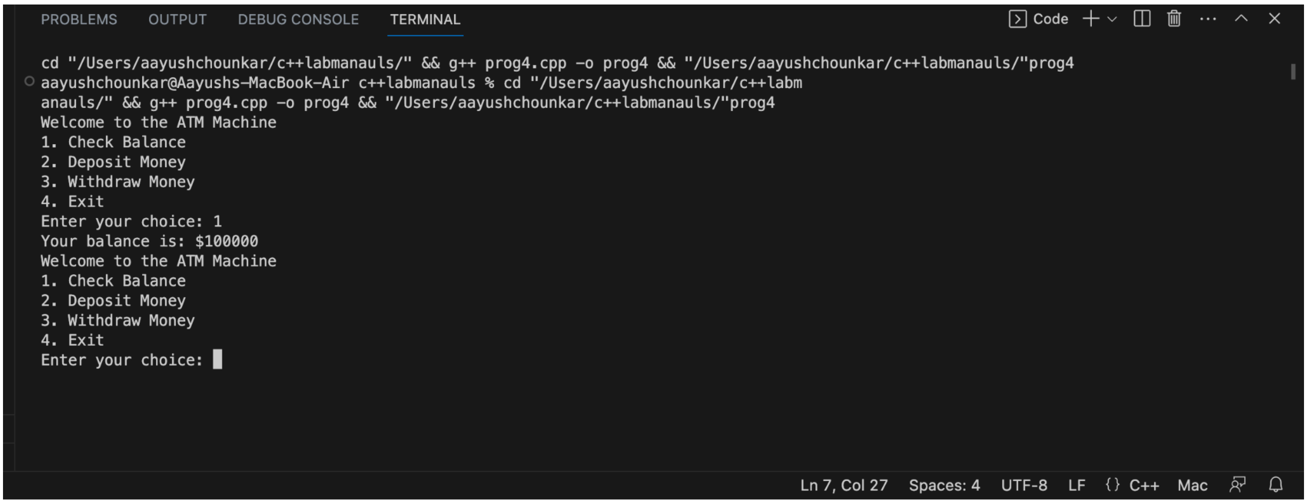
```
// Implement a C++ program that simulates a simple ATM machine, allowing users to check
their balance, deposit, or withdraw money using a switch statement.

#include <iostream>

using namespace std;


int main()

{

   float balance = 10000000.00;

   int choice;

   float amount;


   while (true)

   {

       cout << "Welcome to the ATM Machine" << endl;

       cout << "1. Check Balance" << endl;

       cout << "2. Deposit Money" << endl;

       cout << "3. Withdraw Money" << endl;
```

```
        cout << "4. Exit" << endl;

        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice)
        {
        case 1:
            cout << "Your balance is: $" << balance << endl;
            break;
        case 2:
            cout << "Enter the amount you want to deposit: $";
            cin >> amount;
            balance += amount;
            cout << "Deposit successful. Your new balance is: $" << balance << endl;
            break;
        case 3:
            cout << "Enter the amount you want to withdraw: $";
            cin >> amount;
            if (amount > balance)
            {
                cout << "Insufficient funds!" << endl;
            }
            else
            {
                balance -= amount;
                cout << "Withdrawal successful. Your new balance is: $" << balance << endl;
            }
            break;
        case 4:
```



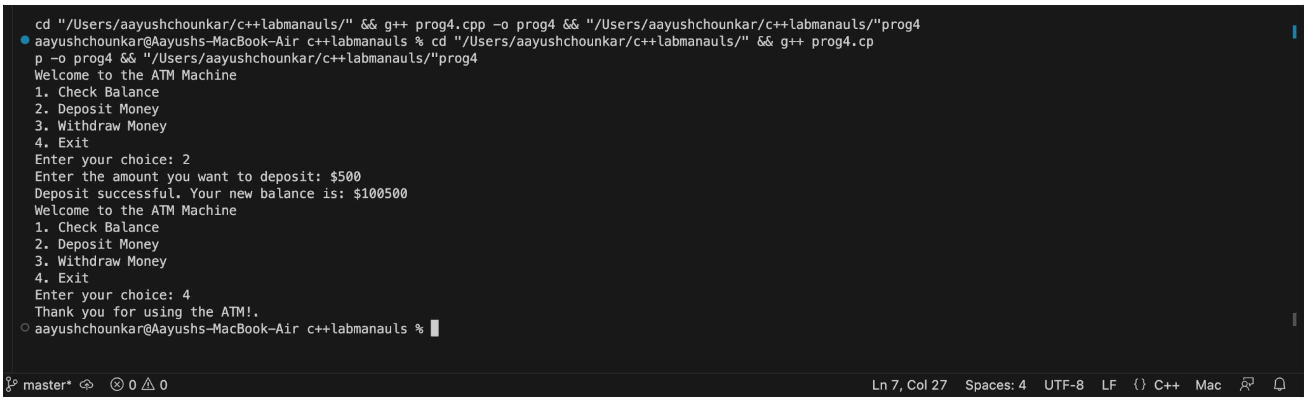
```
            cout << "Thank you for using the ATM!." << endl;

            return 0;

        default:

            cout << "Invalid choice. Please try again." << endl;

            break;

        }

    }

    return 0;

}
```

- OUTPUT-



```
cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog4.cpp -o prog4 && "/Users/aayushchounkar/c++labmanauls/"prog4
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labm
anauls/" && g++ prog4.cpp -o prog4 && "/Users/aayushchounkar/c++labmanauls/"prog4
Welcome to the ATM Machine
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Enter your choice: 1
Your balance is: $100000
Welcome to the ATM Machine
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Enter your choice: 
```

- TEST CASE-



```
cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog4.cpp -o prog4 && "/Users/aayushchounkar/c++labmanauls/"prog4
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog4.cp
p -o prog4 && "/Users/aayushchounkar/c++labmanauls/"prog4
Welcome to the ATM Machine
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Enter your choice: 2
Enter the amount you want to deposit: $500
Deposit successful. Your new balance is: $100500
Welcome to the ATM Machine
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Enter your choice: 4
Thank you for using the ATM!.
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % 
```

- CONCLUSION-
- • The program successfully simulates an ATM, enabling users to check balance, deposit, or withdraw money through a user-friendly switch statement interface.

**Name of Student:Aayush  Chounkar**

**Roll Number: 04**

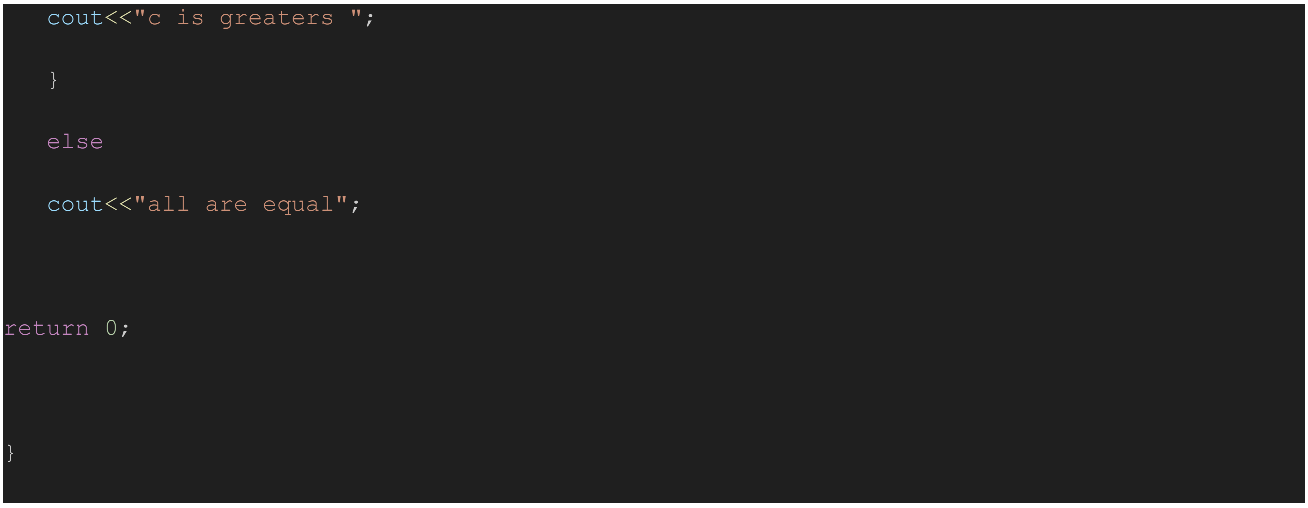
**Experiment No:05**

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

- THEORY-The program is using nested if-else statements to determine the largest number among three
- If three numbers are equal is says all three numbers are equal .

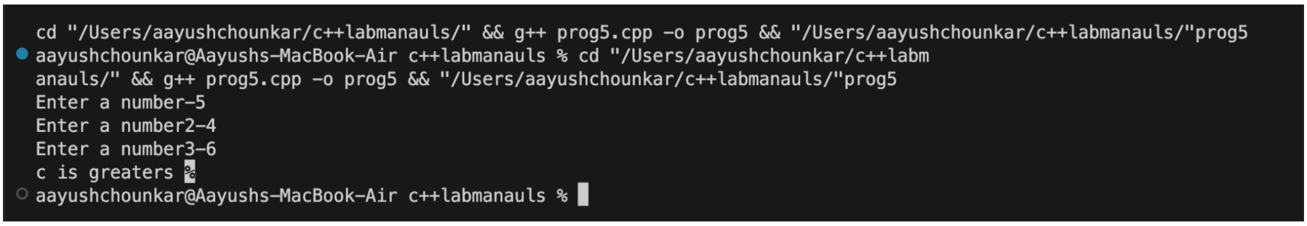
- CODE-

```
#include <iostream>
using namespace std;
int main(){
   int a,b,c;
   cout<<"Enter a number-";
   cin>>a;
   cout<<"Enter a number2-";
   cin>>b;
   cout<<"Enter a number3-";
   cin>>c;
   if (a>b && a>c){
   cout<<"a is greater";
}
   else if (b>a && b>c)
   {
    cout<<"b is greater";
}
   else if (c>a && c>b)
   {
```



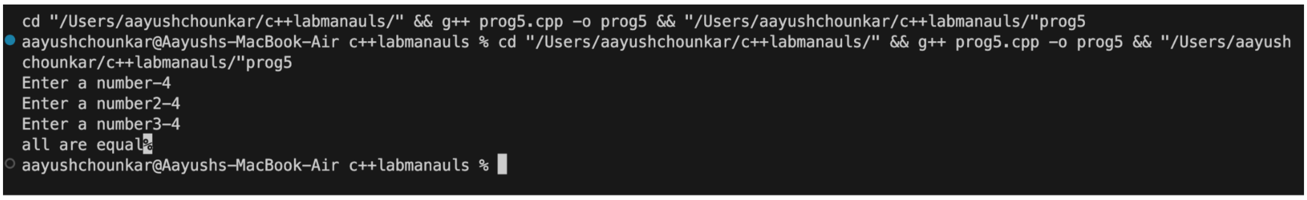
```
    cout<<"c is greaters ";

    }

    else

    cout<<"all are equal";



return 0;



}
```

- OUTPUT-



```
cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog5.cpp -o prog5 && "/Users/aayushchounkar/c++labmanauls/"prog5
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labm
anauls/" && g++ prog5.cpp -o prog5 && "/Users/aayushchounkar/c++labmanauls/"prog5
Enter a number-5
Enter a number2-4
Enter a number3-6
c is greaters %
aayushchounkar@Aayushs-MacBook-Air c++labmanauls %
```

- TESTCASE-



```
cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog5.cpp -o prog5 && "/Users/aayushchounkar/c++labmanauls/"prog5
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog5.cpp -o prog5 && "/Users/aayush
chounkar/c++labmanauls/"prog5
Enter a number-4
Enter a number2-4
Enter a number3-4
all are equal%
aayushchounkar@Aayushs-MacBook-Air c++labmanauls %
```

- CONCLUSION-
- The program is giving accurate result states the biggest number among three and if numbers given are equal it says all numbers are equal

**Name of Student:Aayush  Chounkar**

**Roll Number: 04**

**Experiment No:06**

---

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

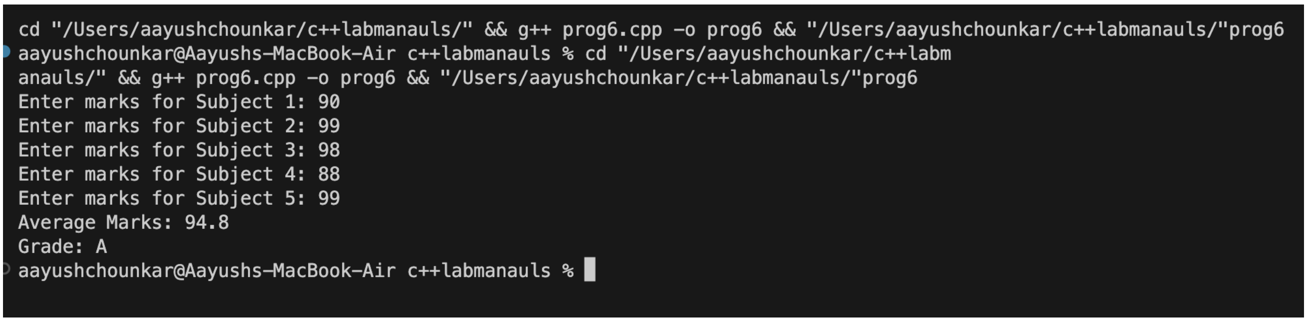
- THEORY-
- The program employs an if-else-if ladder to evaluate a student's grade based on marks in 5 subjects.
- It checks different conditions for various grade ranges, assigning the corresponding grade based on the total marks.

- CODE-

```
#include <iostream>


using namespace std;


int main()

{

   // Get marks for 5 subjects

   double subject1, subject2, subject3, subject4, subject5;


   cout << "Enter marks for Subject 1: ";

   cin >> subject1;


   cout << "Enter marks for Subject 2: ";

   cin >> subject2;

```

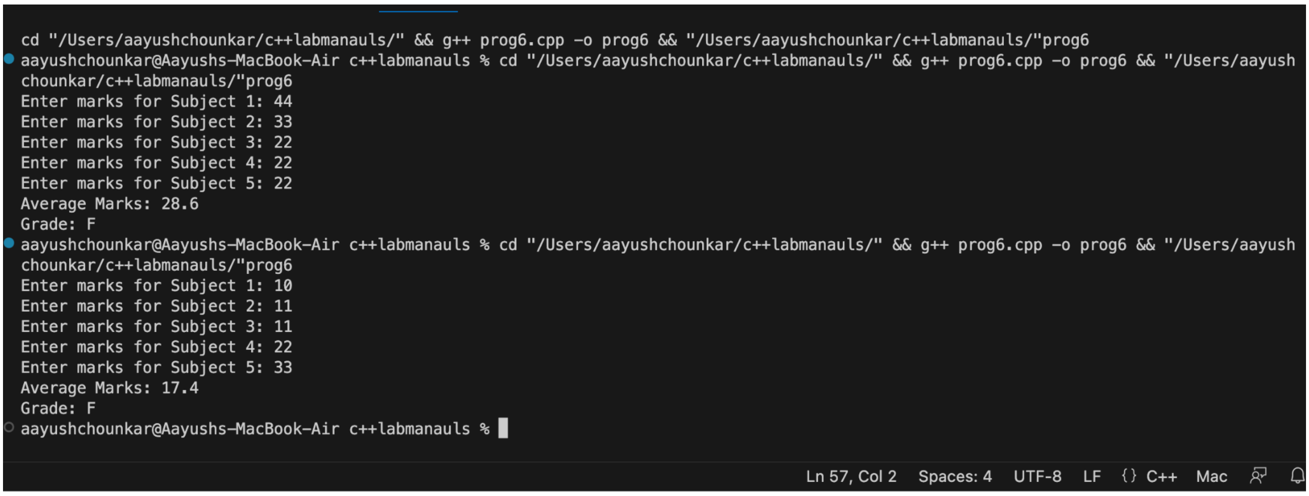
```
cout << "Enter marks for Subject 3: ";
cin >> subject3;

cout << "Enter marks for Subject 4: ";
cin >> subject4;

cout << "Enter marks for Subject 5: ";
cin >> subject5;

// Calculate average marks
double average = (subject1 + subject2 + subject3 + subject4 + subject5) / 5;

// Determine the grade based on average marks
char grade;

if (average >= 90)
{
    grade = 'A';
}
else if (average >= 80)
{
    grade = 'B';
}
else if (average >= 70)
{
    grade = 'C';
}
else if (average >= 60)
{
    grade = 'D';
    }

    else

    {

        grade = 'F';

    }


    // Display the result

    cout << "Average Marks: " << average << endl;

    cout << "Grade: " << grade << endl;


    return 0;

}
```

- OUTPUT-



```
cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog6.cpp -o prog6 && "/Users/aayushchounkar/c++labmanauls/"prog6
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labm
anauls/" && g++ prog6.cpp -o prog6 && "/Users/aayushchounkar/c++labmanauls/"prog6
Enter marks for Subject 1: 90
Enter marks for Subject 2: 99
Enter marks for Subject 3: 98
Enter marks for Subject 4: 88
Enter marks for Subject 5: 99
Average Marks: 94.8
Grade: A
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % 
```

- TESTCASE-



```
cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog6.cpp -o prog6 && "/Users/aayushchounkar/c++labmanauls/"prog6
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog6.cpp -o prog6 && "/Users/aayush
chounkar/c++labmanauls/"prog6
Enter marks for Subject 1: 44
Enter marks for Subject 2: 33
Enter marks for Subject 3: 22
Enter marks for Subject 4: 22
Enter marks for Subject 5: 22
Average Marks: 28.6
Grade: F
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog6.cpp -o prog6 && "/Users/aayush
chounkar/c++labmanauls/"prog6
Enter marks for Subject 1: 10
Enter marks for Subject 2: 11
Enter marks for Subject 3: 11
Enter marks for Subject 4: 22
Enter marks for Subject 5: 33
Average Marks: 17.4
Grade: F
aayushchounkar@Aayushs-MacBook-Air c++labmanauls %
```

- CONCLUSION -
- The program accurately assigns a grade to a student by evaluating the total marks in 5 subjects using an if-else-if ladder.

**Name of Student:Aayush  Chounkar**

**Roll Number: 04**

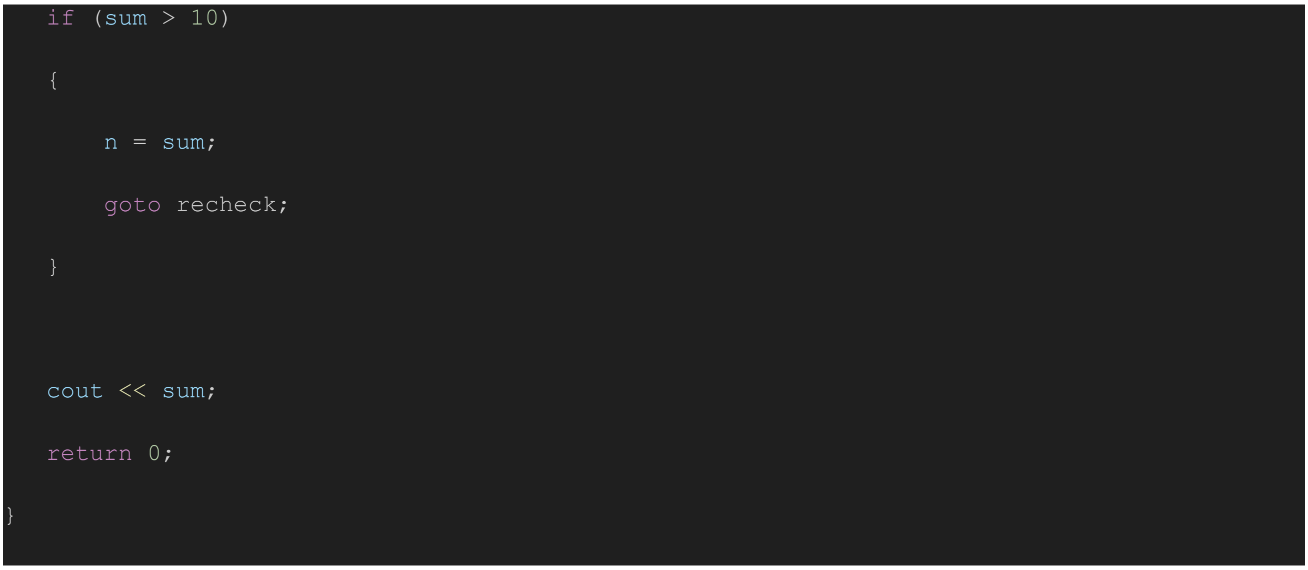
**Experiment No:07**

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

- THEORY-
- The program uses a loop to repeatedly sum the digits of a number until it becomes a single-digit number.
- It extracts and adds each digit in the loop until the sum is

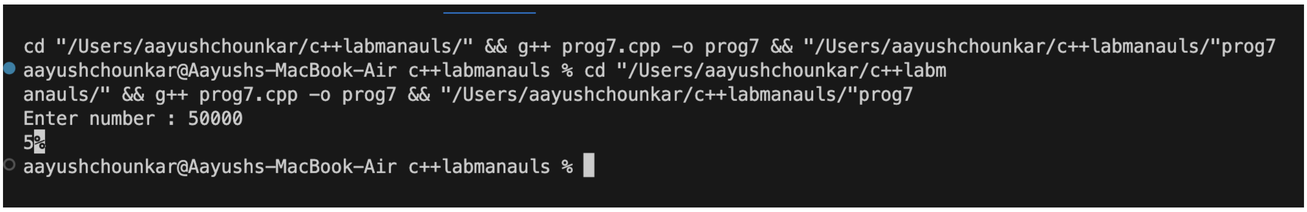
- CODE-

```
#include <iostream>
using namespace std;
int main()
{
   int n, digit, sum;

   cout << "Enter number : ";
   cin >> n;
recheck:
   sum = 0;
   while (n > 0)
   {
       digit = n % 10;
       sum += digit;
       n /= 10;
   }
```



```
    if (sum > 10)
    {
        n = sum;
        goto recheck;
    }

    cout << sum;
    return 0;
}
```

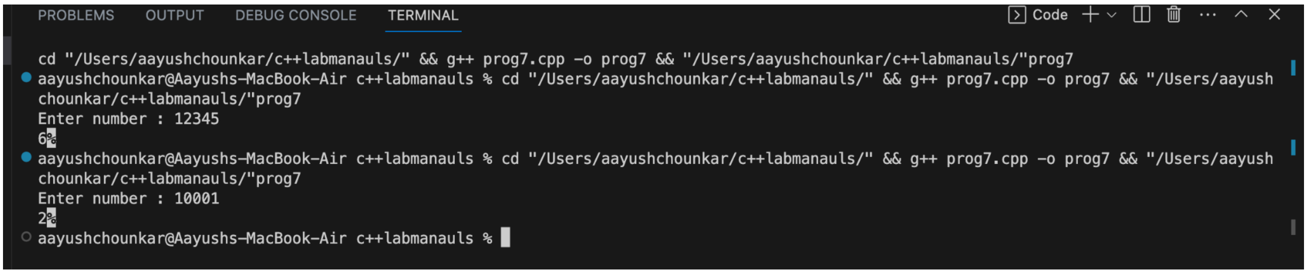
- OUTPUT-



```
cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog7.cpp -o prog7 && "/Users/aayushchounkar/c++labmanauls/"prog7
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labm
anauls/" && g++ prog7.cpp -o prog7 && "/Users/aayushchounkar/c++labmanauls/"prog7
Enter number : 50000
5%
aayushchounkar@Aayushs-MacBook-Air c++labmanauls %
```

-

- TESTCASE-



```
cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog7.cpp -o prog7 && "/Users/aayushchounkar/c++labmanauls/"prog7
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog7.cpp -o prog7 && "/Users/aayush
chounkar/c++labmanauls/"prog7
Enter number : 12345
6%
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog7.cpp -o prog7 && "/Users/aayush
chounkar/c++labmanauls/"prog7
Enter number : 10001
2%
aayushchounkar@Aayushs-MacBook-Air c++labmanauls %
```

- CONCLUSION-
- The program sucessfully calculates sum of number until it becomes a singl digit number.

**Name of Student:Aayush  Chounkar**

**Roll Number: 04**

**Experiment No:08**

- TITLE- Write a program to print a Pascal's triangle using nested loops.

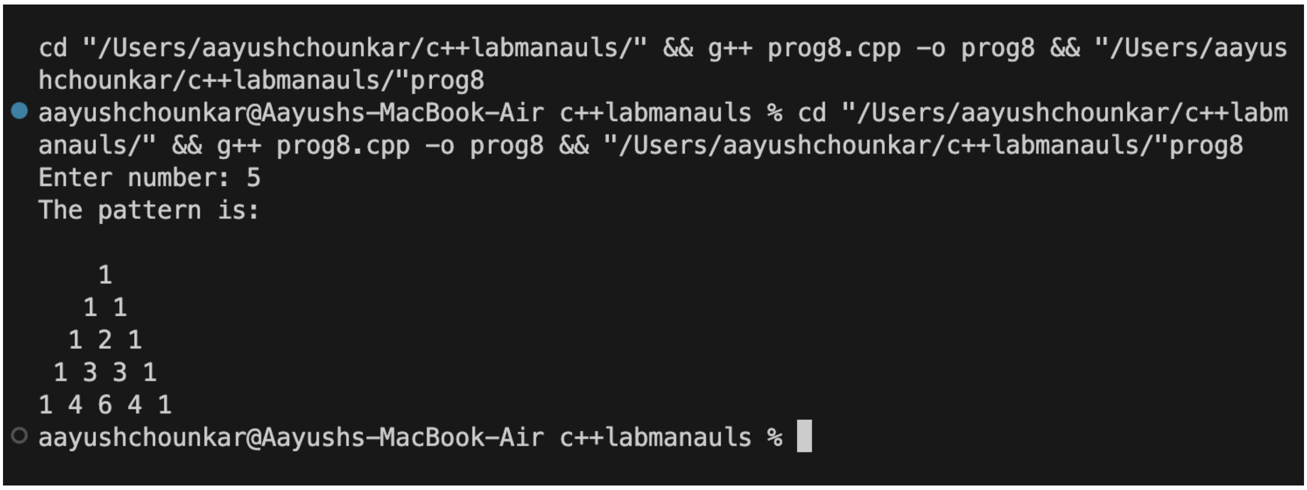
- THEORY-
- Pascal's triangle is formed by summing adjacent elements from the row above.
- The program uses nested loops to calculate and print the coefficients, arranging them in a triangular pattern.

- 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;

}
```

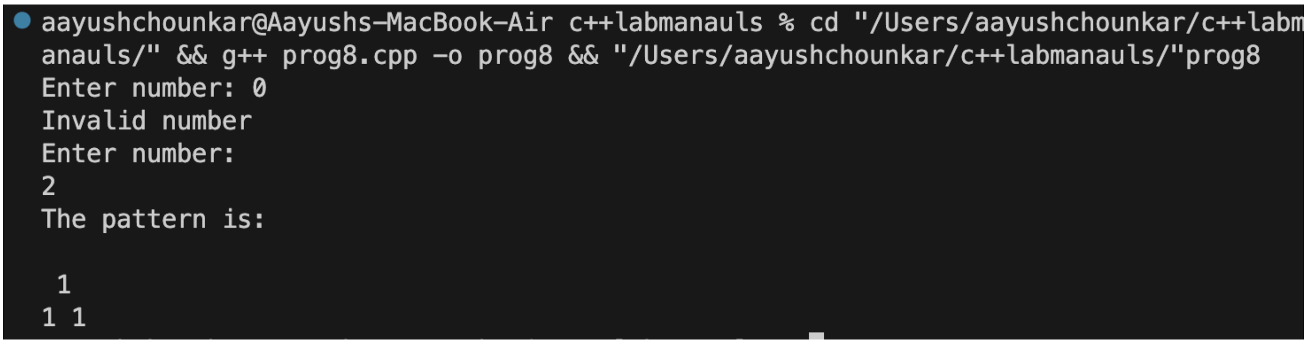
- OUTPUT-



```
cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog8.cpp -o prog8 && "/Users/aayushchounkar/c++labmanauls/"prog8
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog8.cpp -o prog8 && "/Users/aayushchounkar/c++labmanauls/"prog8
Enter number: 5
The pattern is:

    1
   1 1
  1 2 1
 1 3 3 1
1 4 6 4 1
aayushchounkar@Aayushs-MacBook-Air c++labmanauls %
```

-

- TESTCASE-

- 

```
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labm
anauls/" && g++ prog8.cpp -o prog8 && "/Users/aayushchounkar/c++labmanauls/"prog8
Enter number: 0
Invalid number
Enter number:
2
The pattern is:

 1
1 1
```

- CONCLUSION-
- The program effectively prints Pascal's triangle using nested loops, showcasing the pattern of coefficients formed by adding adjacent elements from the row above.

**Name of Student:Aayush  Chounkar**

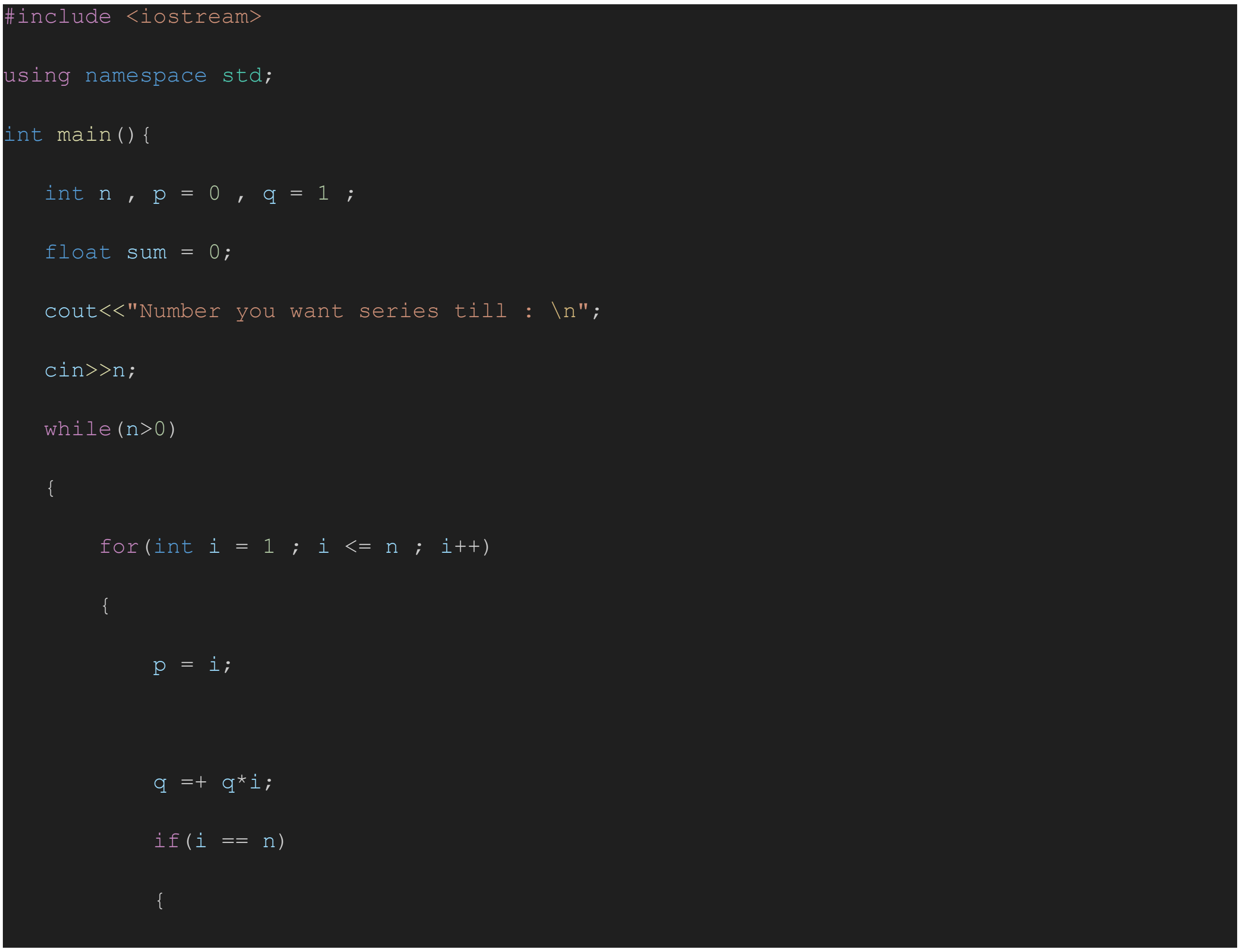
**Roll Number: 04**

**Experiment No:09**

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

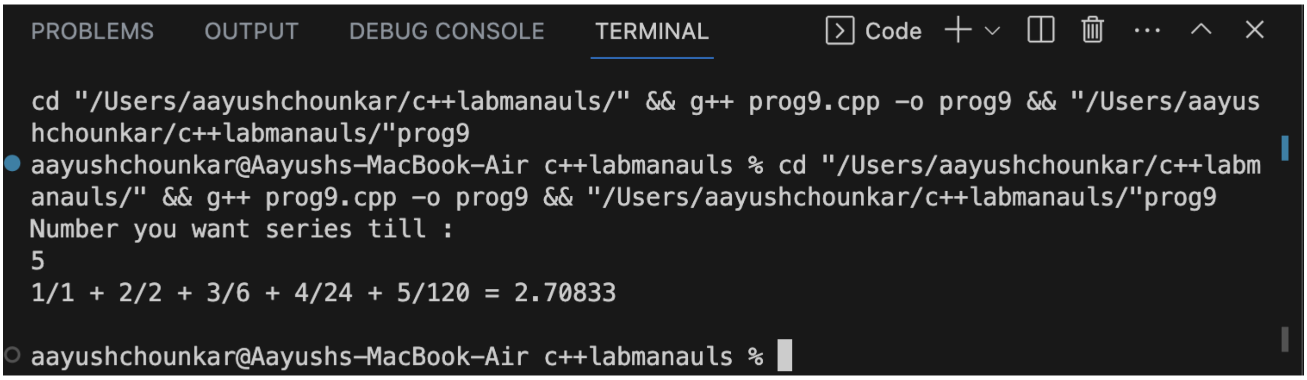
- THEORY-
- The program uses nested loops to calculate the sum of the series 1/1! + 2/2! + 3/3! + ... + N/N!.
- Nested loops compute each term of the series, where the inner loop calculates the factorial.

- CODE-



```
#include <iostream>

using namespace std;

int main(){

   int n , p = 0 , q = 1 ;

   float sum = 0;

   cout<<"Number you want series till : \n";

   cin>>n;

   while(n>0)

   {

       for(int i = 1 ; i <= n ; i++)

       {

           p = i;


           q =+ q*i;

           if(i == n)

           {
                cout<<p<<"/"<<q<<" = "<<sum+(float(p)/float(q))<<"\n\n";

                return 0;

            }

            else

            {

                cout<<p<<"/"<<q<<" + ";

                sum += float(p)/float(q);

            }

        }

    }

    cout<<"Invalid input please put positive number greater than zero ..Thank you\n\n";

    return 0;

}
```

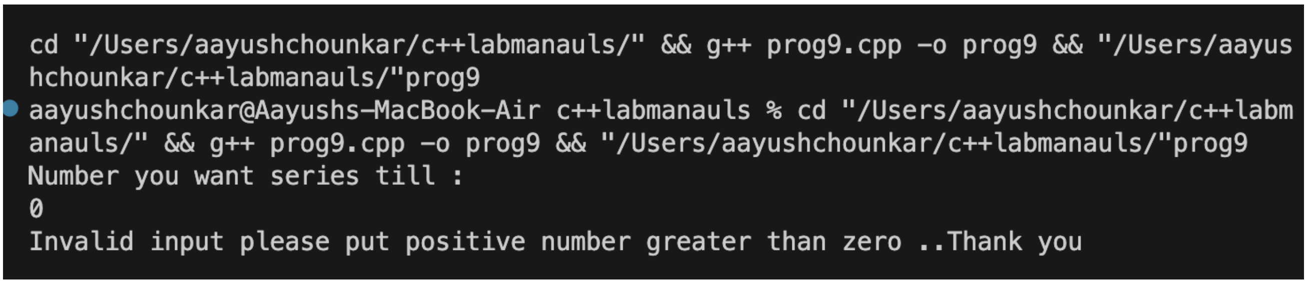
- OUTPUT-



```
cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog9.cpp -o prog9 && "/Users/aayus
hchounkar/c++labmanauls/"prog9
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labm
anauls/" && g++ prog9.cpp -o prog9 && "/Users/aayushchounkar/c++labmanauls/"prog9
Number you want series till :
5
1/1 + 2/2 + 3/6 + 4/24 + 5/120 = 2.70833

aayushchounkar@Aayushs-MacBook-Air c++labmanauls %
```

-

- TESTCASE



```
cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog9.cpp -o prog9 && "/Users/aayus
hchounkar/c++labmanauls/"prog9
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labm
anauls/" && g++ prog9.cpp -o prog9 && "/Users/aayushchounkar/c++labmanauls/"prog9
Number you want series till :
0
Invalid input please put positive number greater than zero ..Thank you
```

- CONCLUSION-
- The program accurately calculates and displays the sum of the series 1/1! + 2/2! + 3/3! + ... + N/N! using nested loops

---

**Name of Student:Aayush  Chounkar**

**Roll Number: 04**

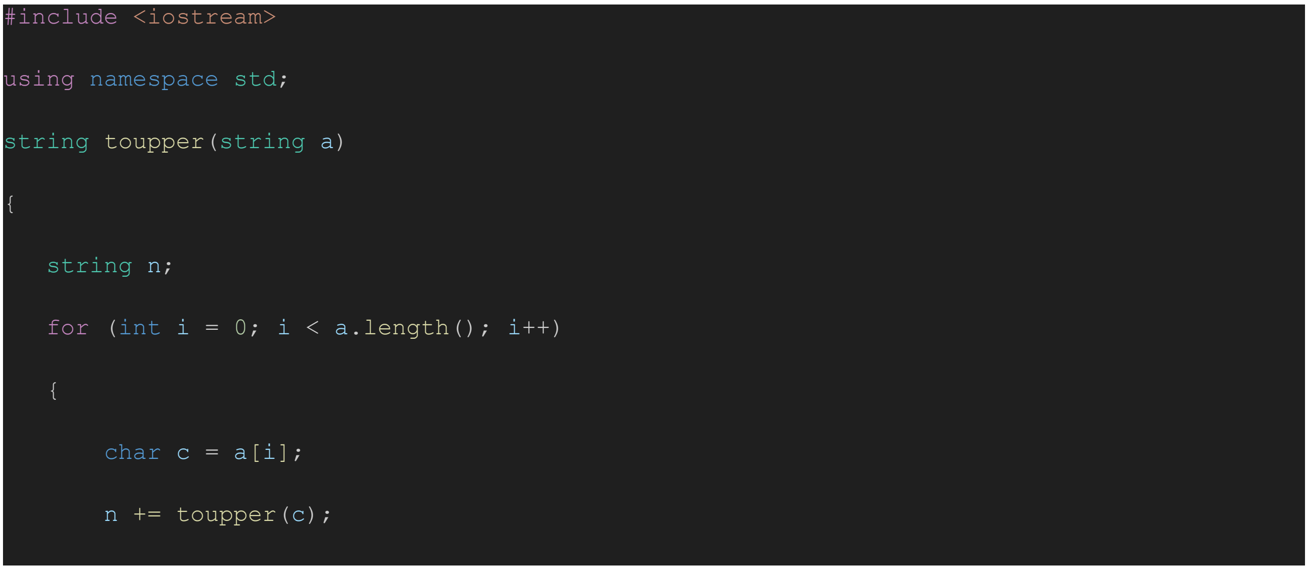
**Experiment No:10**

---

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

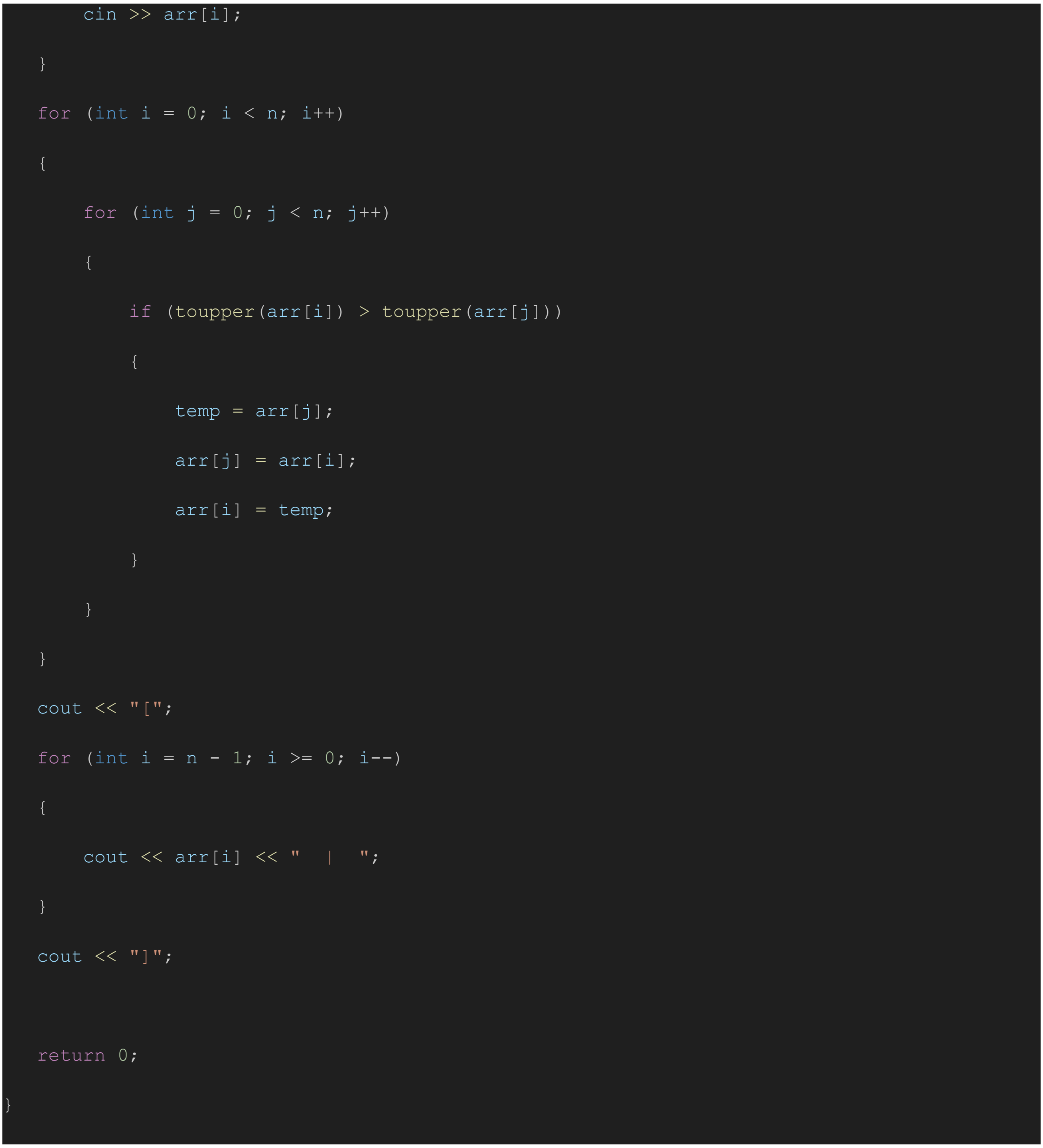
- THEORY-
- The program creates an array of strings and sorts them alphabetically using a sorting algorithm.
- It arranges the strings in ascending order and displays the sorted array

- CODE-



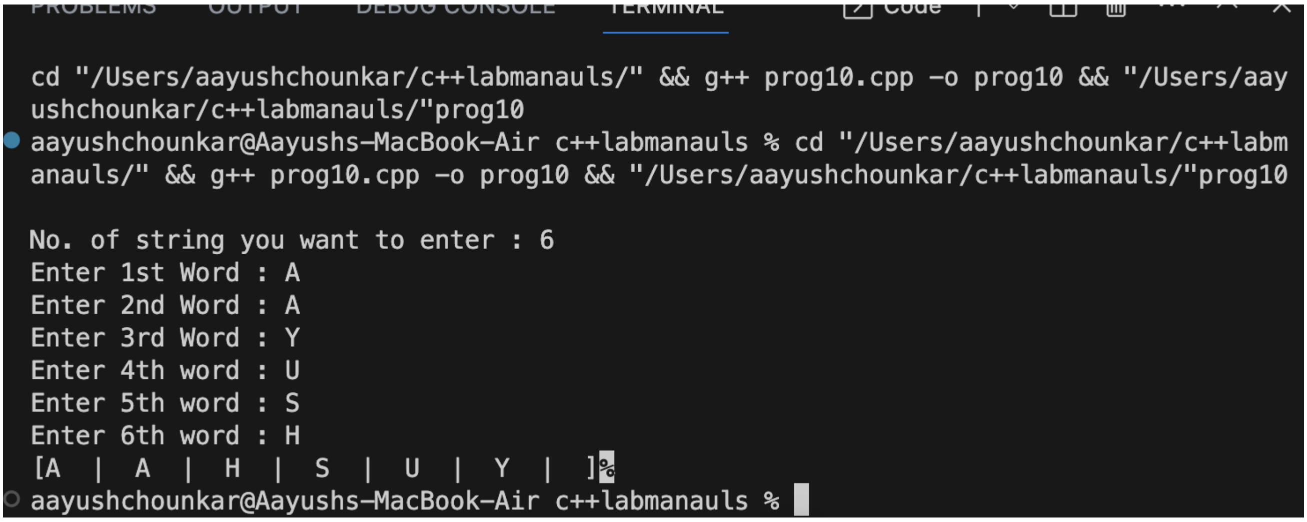
```
#include <iostream>

using namespace std;

string toupper(string a)

{

   string n;

   for (int i = 0; i < a.length(); i++)

   {

       char c = a[i];

       n += toupper(c);
```

```
    }
    return n;
}
int main()
{
    int n;
    string temp;
    cout << "No. of string you want to enter : ";
    cin >> n;
    string arr[n];
    for (int i = 0; i < n; i++)
    {
        switch (i)
        {
        case 0:
            cout << "Enter 1st Word : ";
            cin >> arr[i];
            continue;
        case 1:
            cout << "Enter 2nd Word : ";
            cin >> arr[i];
            continue;
        case 2:
            cout << "Enter 3rd Word : ";
            cin >> arr[i];
            continue;
        default:
            break;
        }
        cout << "Enter " << i + 1 << "th word : ";
```



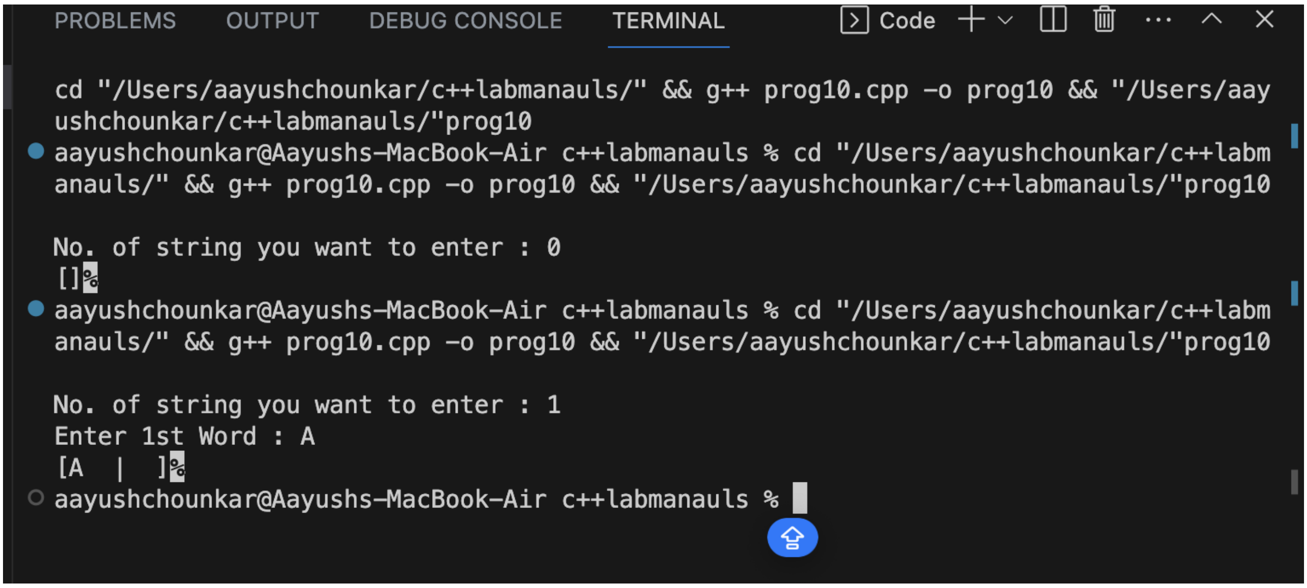
```
        cin >> arr[i];
    }
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            if (toupper(arr[i]) > toupper(arr[j]))
            {
                temp = arr[j];
                arr[j] = arr[i];
                arr[i] = temp;
            }
        }
    }
    cout << "[";
    for (int i = n - 1; i >= 0; i--)
    {
        cout << arr[i] << "  |  ";
    }
    cout << "]";

    return 0;
}
```

- OUTPUT



```
cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog10.cpp -o prog10 && "/Users/aay
ushchounkar/c++labmanauls/"prog10
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labm
anauls/" && g++ prog10.cpp -o prog10 && "/Users/aayushchounkar/c++labmanauls/"prog10

No. of string you want to enter : 6
Enter 1st Word : A
Enter 2nd Word : A
Enter 3rd Word : Y
Enter 4th word : U
Enter 5th word : S
Enter 6th word : H
[A  |  A  |  H  |  S  |  U  |  Y  |  ]%
aayushchounkar@Aayushs-MacBook-Air c++labmanauls %
```

- TESTCASE



```
cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog10.cpp -o prog10 && "/Users/aay
ushchounkar/c++labmanauls/"prog10
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labm
anauls/" && g++ prog10.cpp -o prog10 && "/Users/aayushchounkar/c++labmanauls/"prog10

No. of string you want to enter : 0
[]%
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labm
anauls/" && g++ prog10.cpp -o prog10 && "/Users/aayushchounkar/c++labmanauls/"prog10

No. of string you want to enter : 1
Enter 1st Word : A
[A  |  ]%
aayushchounkar@Aayushs-MacBook-Air c++labmanauls %
```

- CONCLUSION-
- The program successfully creates an array of strings and displays them in alphabetical order through a sorting algorithm, ensuring a clear and organized output.

**Name of Student:Aayush  Chounkar**

**Roll Number: 04**

**Experiment No:11**

- TITLE-Write a program that checks if an array is sorted in ascending order.

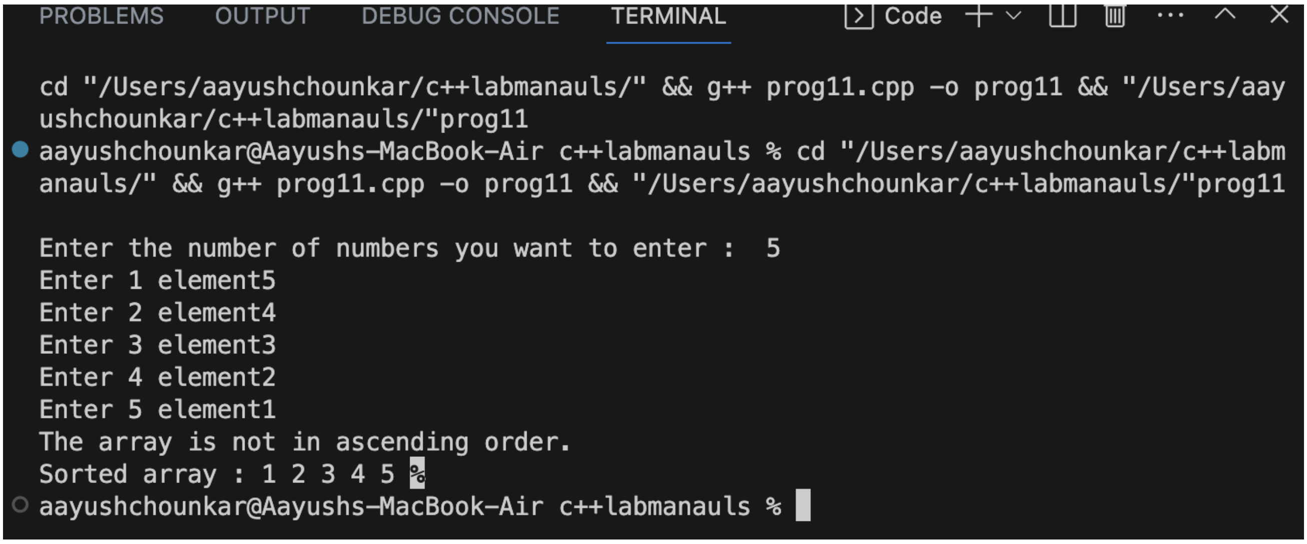
- THE0RY-
- The program checks if an array is sorted in ascending order by comparing each element with the next one. It iterates through the array, verifying that each element is less than or equal to the next

- CODE

```
#include <iostream>
using namespace std;
int main()
{
   float temp;
   int n;
   bool check;
   cout << "Enter the number of numbers you want to enter :  ";
   cin >> n;
   float arr[n];
   for (int i = 0; i < n; i++)
   {
       cout << "Enter " << i + 1 << " element";
       cin >> arr[i];
   }
   for (int i = 0; i < n - 1; i++)
   {
    if (arr[i] > arr[i + 1])

    {

        check = false;

    }

}

if (check == false)

{

    cout << "The array is not in ascending order.\nSorted array : ";

}

else

{

    cout << "Array is in ascending order ";

    return 0;

}

for (int i = 0; i < n; i++)

{

    for (int j = 0; j < n; j++)

    {

        if (arr[i] < arr[j])

        {

            temp = arr[i];

            arr[i] = arr[j];

            arr[j] = temp;

        }

    }

}

for (int i = 0; i < n; i++)

{

    cout << arr[i] << " ";

}
```

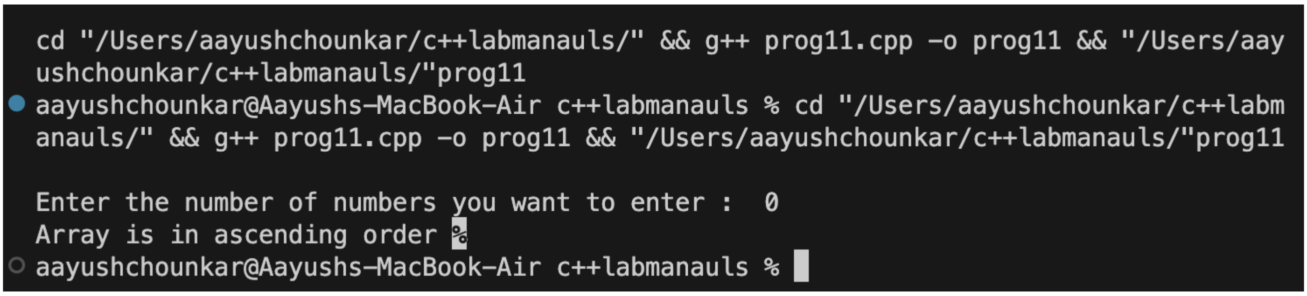
```
    return 0;

}
```

- OUTPUT



```
cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog11.cpp -o prog11 && "/Users/aay
ushchounkar/c++labmanauls/"prog11
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labm
anauls/" && g++ prog11.cpp -o prog11 && "/Users/aayushchounkar/c++labmanauls/"prog11

Enter the number of numbers you want to enter :  5
Enter 1 element5
Enter 2 element4
Enter 3 element3
Enter 4 element2
Enter 5 element1
The array is not in ascending order.
Sorted array : 1 2 3 4 5 %
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % 
```

- TESTCASE-



```
cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog11.cpp -o prog11 && "/Users/aay
ushchounkar/c++labmanauls/"prog11
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labm
anauls/" && g++ prog11.cpp -o prog11 && "/Users/aayushchounkar/c++labmanauls/"prog11

Enter the number of numbers you want to enter :  0
Array is in ascending order %
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % 
```

- CONCLUSION

The program accurately determines whether the given array is sorted in ascending order by comparing each element with the succeeding one.

**Name of Student:Aayush  Chounkar**

**Roll Number: 04**

**Experiment No:12**

---

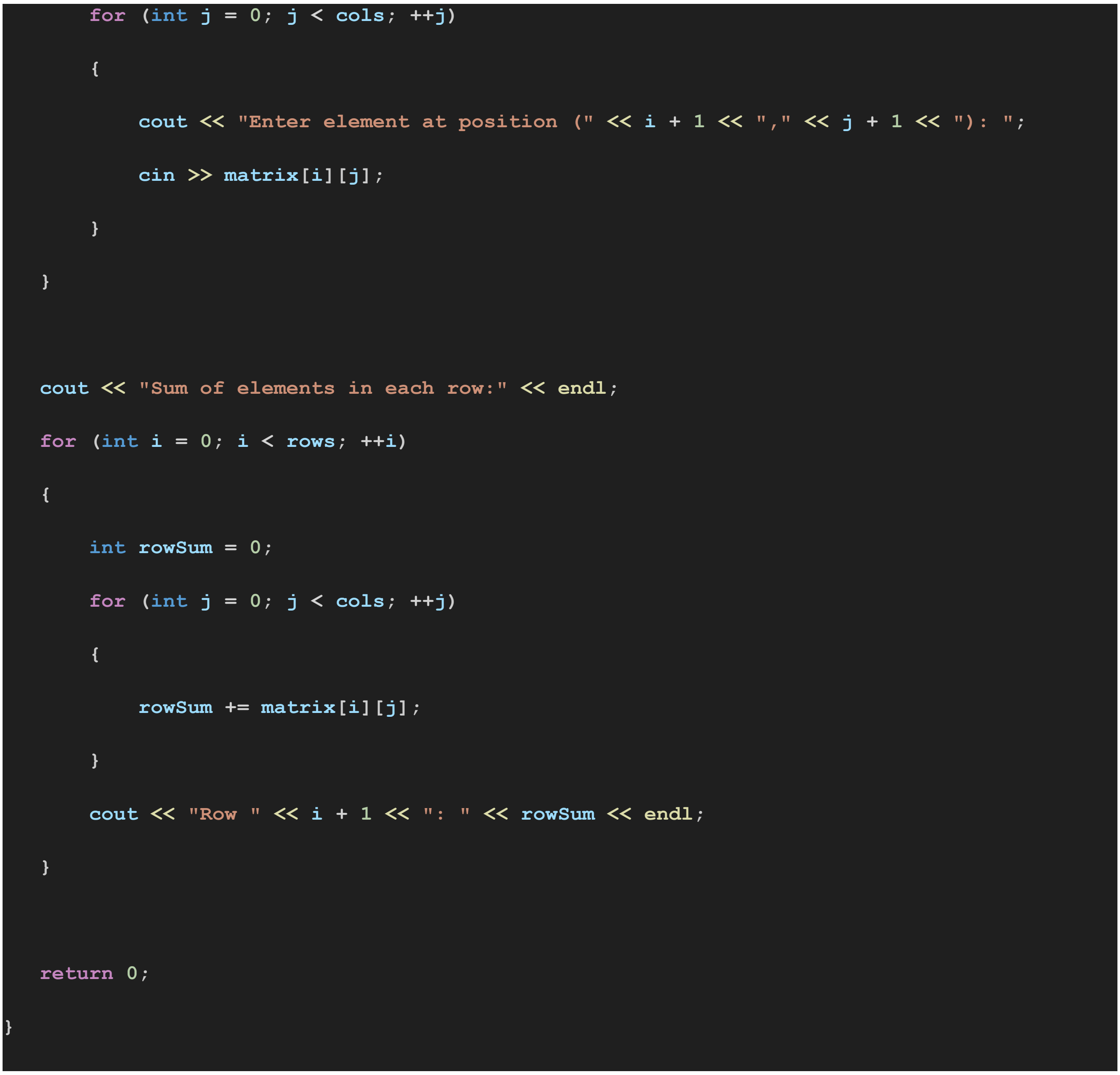
**TITLE-**Write a program to calculate the sum of elements in each row of a matrix.

**THEORY-**

**The program calculates the sum of elements in each row of a matrix using nested loops. It iterates through each row, adding up the elements, and displays the sum for each row**

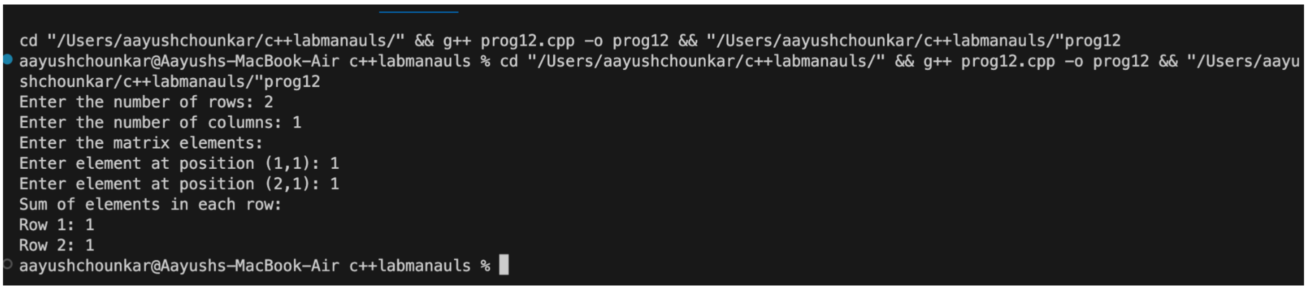
**CODE-**

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

int main()
{
   int rows, cols;

   cout << "Enter the number of rows: ";
   cin >> rows;

   cout << "Enter the number of columns: ";
   cin >> cols;

   int matrix[rows][cols];

   cout << "Enter the matrix elements:" << endl;
   for (int i = 0; i < rows; ++i)
   {
```



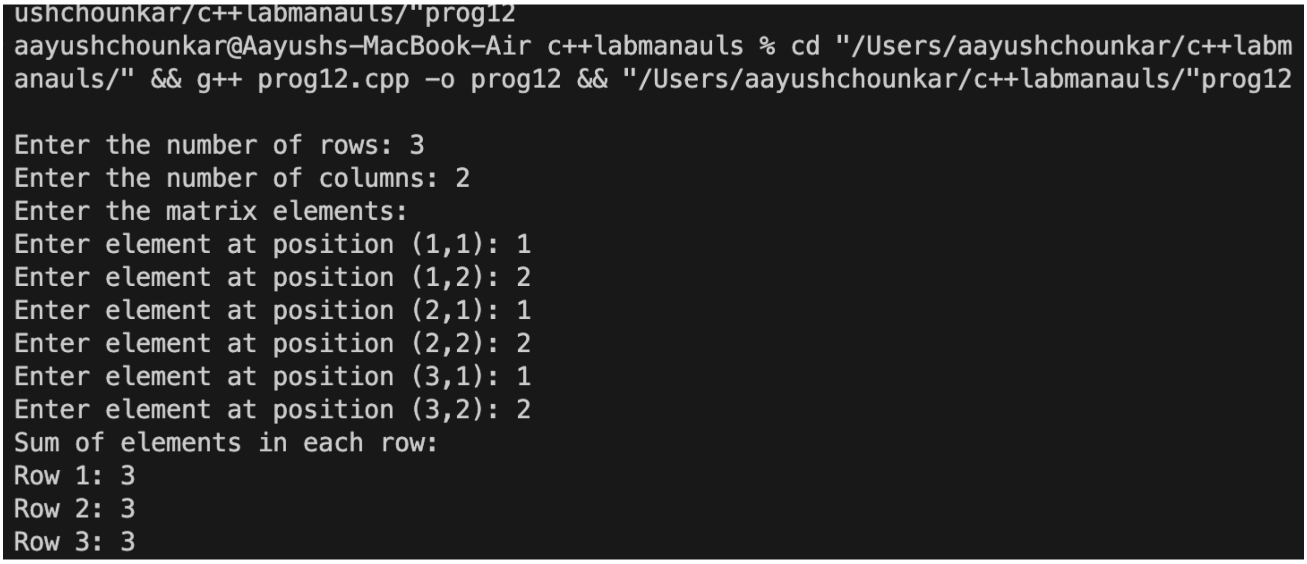
```
        for (int j = 0; j < cols; ++j)
        {
            cout << "Enter element at position (" << i + 1 << "," << j + 1 << "): ";
            cin >> matrix[i][j];
        }
    }

    cout << "Sum of elements in each row:" << endl;
    for (int i = 0; i < rows; ++i)
    {
        int rowSum = 0;
        for (int j = 0; j < cols; ++j)
        {
            rowSum += matrix[i][j];
        }
        cout << "Row " << i + 1 << ": " << rowSum << endl;
    }

    return 0;
}
```

# OUTPUT



```
cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog12.cpp -o prog12 && "/Users/aayushchounkar/c++labmanauls/"prog12
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog12.cpp -o prog12 && "/Users/aayu
shchounkar/c++labmanauls/"prog12
Enter the number of rows: 2
Enter the number of columns: 1
Enter the matrix elements:
Enter element at position (1,1): 1
Enter element at position (2,1): 1
Sum of elements in each row:
Row 1: 1
Row 2: 1
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % 
```

## TESTCASE



```
ushchounkar/c++labmanauls/"prog12
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labm
anauls/" && g++ prog12.cpp -o prog12 && "/Users/aayushchounkar/c++labmanauls/"prog12

Enter the number of rows: 3
Enter the number of columns: 2
Enter the matrix elements:
Enter element at position (1,1): 1
Enter element at position (1,2): 2
Enter element at position (2,1): 1
Enter element at position (2,2): 2
Enter element at position (3,1): 1
Enter element at position (3,2): 2
Sum of elements in each row:
Row 1: 3
Row 2: 3
Row 3: 3
```

## CONCLUSION

**The program successfully calculates and displays the sum of elements in each row of a matrix, providing a clear and organized output**

**Name of Student:Aayush  Chounkar**

**Roll Number: 04**

**Experiment No:13**

---

**TITLE-**Write a program to generate all possible permutations of a string.

THEORY

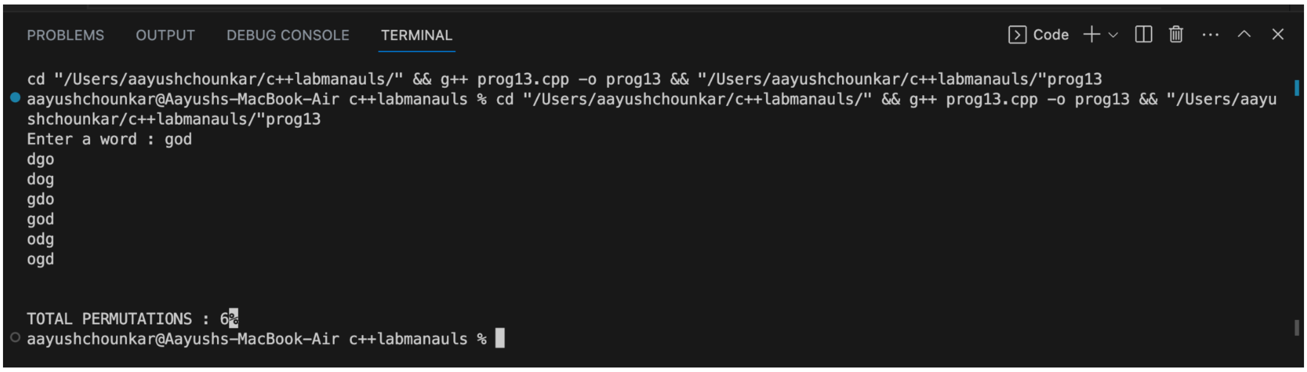
The program generates all possible permutations of a string using recursive backtracking.

- It uses a function that allows to check for the next permutation

## CODE

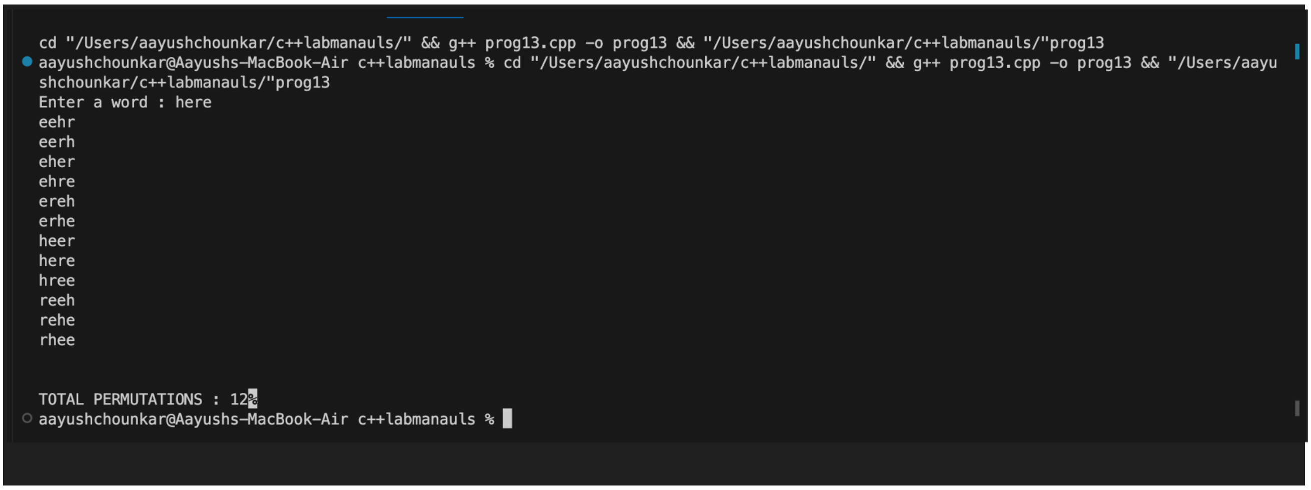
```
#include <iostream>

using namespace std;

int main() {

int a = 0;

string input_string;

cout<<"Enter a word : ";

cin>>input_string;

sort(input_string.begin(), input_string.end());

do {

cout << input_string << endl;

a++;

} while (next_permutation(input_string.begin(),

input_string.end()));

cout<<"\n\nTOTAL PERMUTATIONS : "<<a;

return 0;

}
```

## OUTPUT



```
cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog13.cpp -o prog13 && "/Users/aayushchounkar/c++labmanauls/"prog13
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog13.cpp -o prog13 && "/Users/aayu
shchounkar/c++labmanauls/"prog13
Enter a word : god
dgo
dog
gdo
god
odg
ogd


TOTAL PERMUTATIONS : 6%
aayushchounkar@Aayushs-MacBook-Air c++labmanauls %
```

## TESTCASE



```
cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog13.cpp -o prog13 && "/Users/aayushchounkar/c++labmanauls/"prog13
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog13.cpp -o prog13 && "/Users/aayu
shchounkar/c++labmanauls/"prog13
Enter a word : here
eehr
eerh
eher
ehre
ereh
erhe
heer
here
hree
reeh
rehe
rhee


TOTAL PERMUTATIONS : 12%
aayushchounkar@Aayushs-MacBook-Air c++labmanauls %
```

## CONCLUSION

**The program accurately generates all possible permutations of a string, systematically exploring and displaying each combination.**

**Name of Student:Aayush  Chounkar**

**Roll Number: 04**

**Experiment No:14**

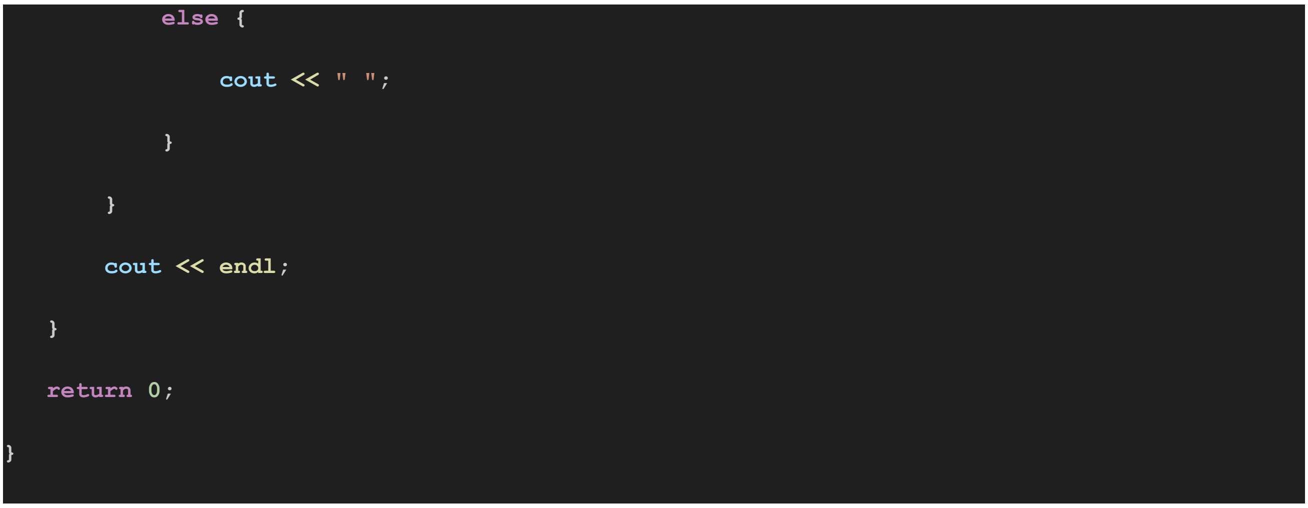
**TITLE-**Create a C++ program to print the following pattern:

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

THEORY-
The pattern consists of five rows and five columns. • '*' is printed in the first and last row, and first and last column, while spaces are printed in the interior

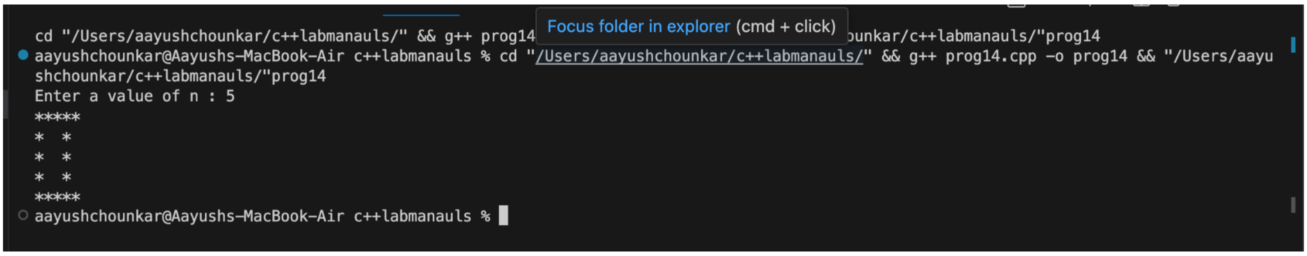
## CODE

```
#include<iostream>

using namespace std;


int main() {

   int n ;

   cout << "Enter a value of n : ";

   cin >> n;


   for(int i = 1; i <= n; i++) {

       for(int j = 1; j <= n; j++) {

           if(i == 1 || i == n|| j == 1 || j == (n-1)) {

               cout << "*";

           }
```



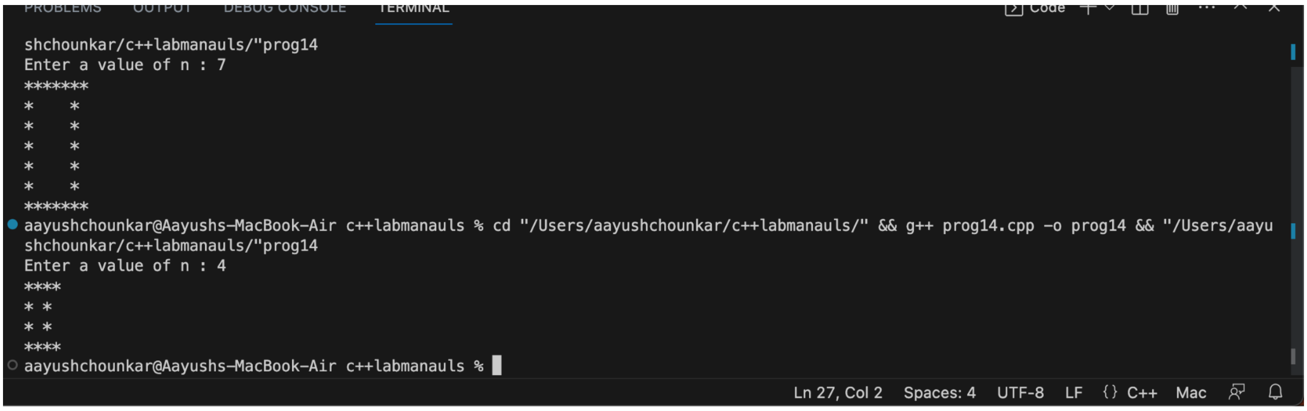
```
            else {

                cout << " ";

            }

        }

        cout << endl;

    }

    return 0;

}
```

# OUTPUT



```
cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog14                                  ounkar/c++labmanauls/"prog14
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog14.cpp -o prog14 && "/Users/aayu
shchounkar/c++labmanauls/"prog14
Enter a value of n : 5
*****
*  *
*  *
*  *
*****
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % 
```

# TESTCASE



```
shchounkar/c++labmanauls/"prog14
Enter a value of n : 7
*******
*    *
*    *
*    *
*    *
*    *
*******
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog14.cpp -o prog14 && "/Users/aayu
shchounkar/c++labmanauls/"prog14
Enter a value of n : 4
****
* *
* *
****
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % 
```

**CONCLUSION**

**This C++ program successfully creates the desired pattern with a combination of '*' and spaces, following the specified row and column structure**

---

---

**Name of Student:Aayush  Chounkar**

**Roll Number: 04**

**Experiment No:15**

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

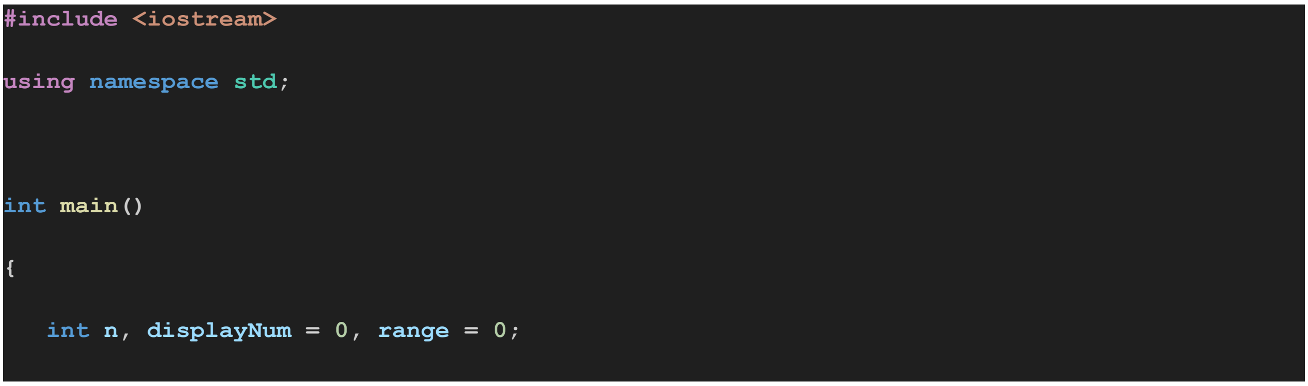
**1**
**232**
**34543**
**4567654**
**34543**
**232**

## THEORY-

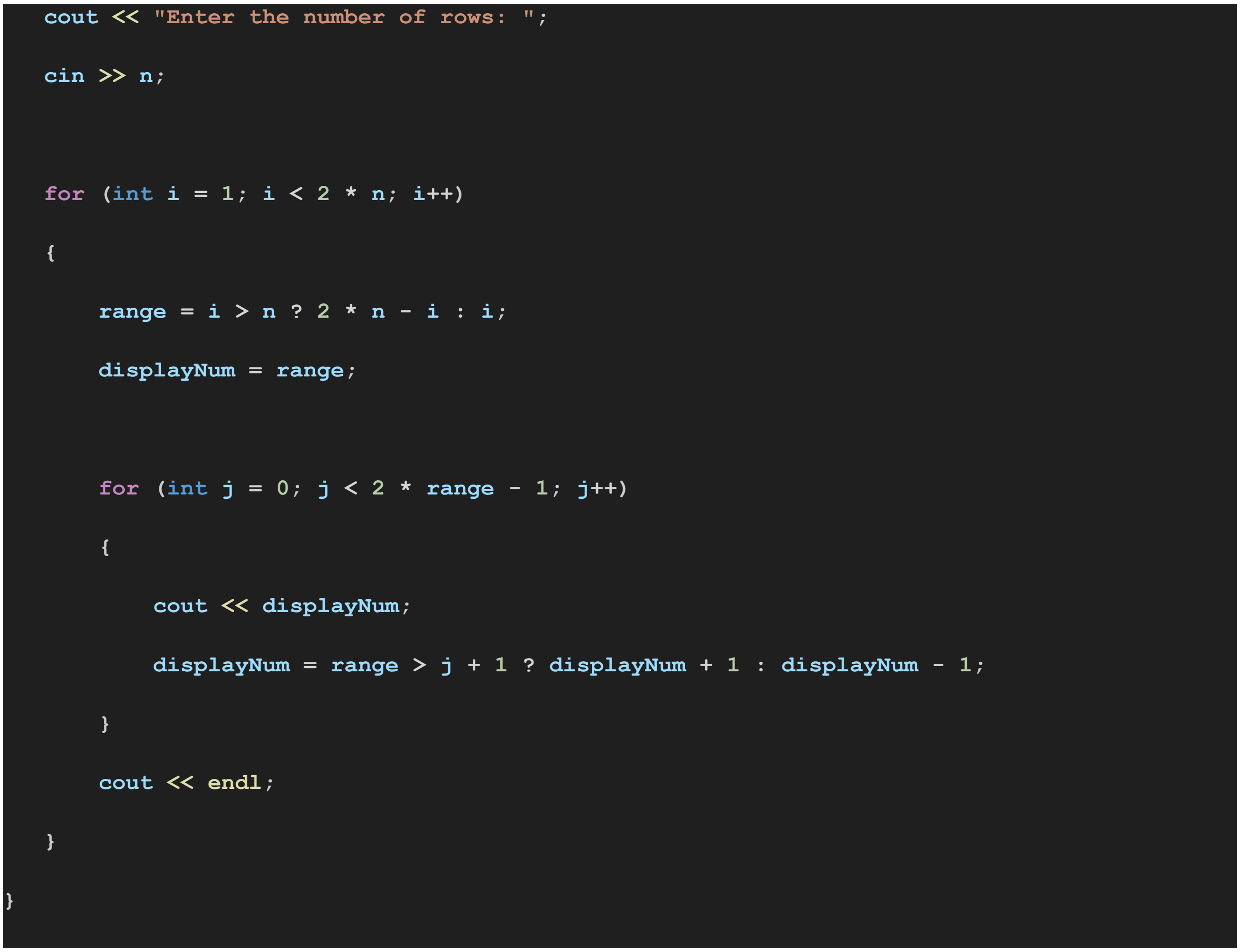
**• The pattern has a symmetric structure with numbers increasing in the first half and decreasing in the second half.**

**• Each row has numbers based on its position in the pattern.**

**CODE**

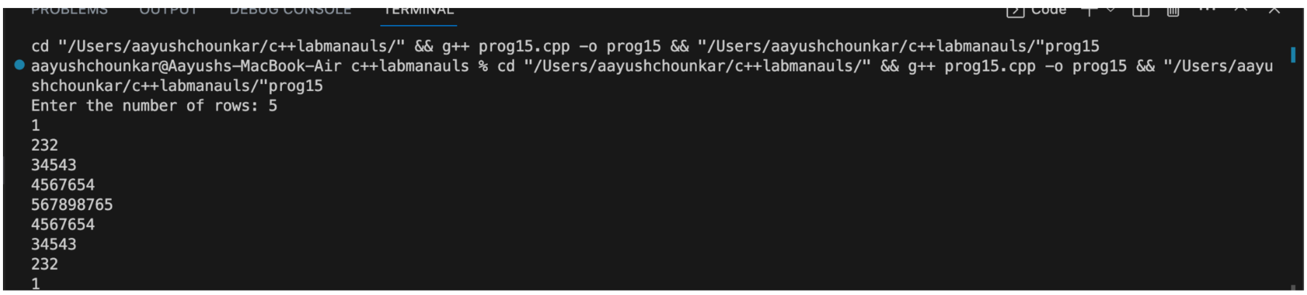


```
#include <iostream>

using namespace std;


int main()

{

    int n, displayNum = 0, range = 0;
```



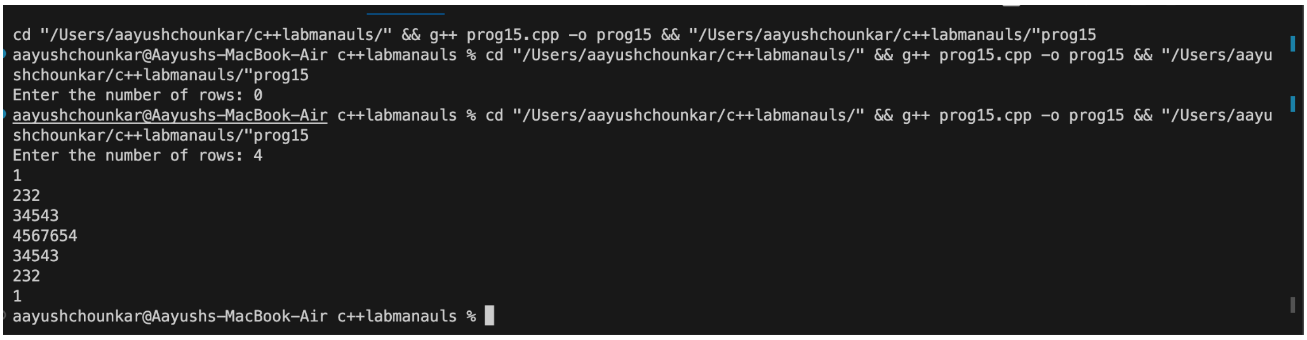
```
    cout << "Enter the number of rows: ";
    cin >> n;

    for (int i = 1; i < 2 * n; i++)
    {
        range = i > n ? 2 * n - i : i;
        displayNum = range;

        for (int j = 0; j < 2 * range - 1; j++)
        {
            cout << displayNum;
            displayNum = range > j + 1 ? displayNum + 1 : displayNum - 1;
        }
        cout << endl;
    }
}
```

## OUTPUT



```
cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog15.cpp -o prog15 && "/Users/aayushchounkar/c++labmanauls/"prog15
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog15.cpp -o prog15 && "/Users/aayu
shchounkar/c++labmanauls/"prog15
Enter the number of rows: 5
1
232
34543
4567654
567898765
4567654
34543
232
1
```

## TESTCASE



```
cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog15.cpp -o prog15 && "/Users/aayushchounkar/c++labmanauls/"prog15
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog15.cpp -o prog15 && "/Users/aayu
shchounkar/c++labmanauls/"prog15
Enter the number of rows: 0
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog15.cpp -o prog15 && "/Users/aayu
shchounkar/c++labmanauls/"prog15
Enter the number of rows: 4
1
232
34543
4567654
34543
232
1
aayushchounkar@Aayushs-MacBook-Air c++labmanauls %
```

## CONCLUSION

**This C++ program generates the requested pattern, with numbers increasing and then decreasing in a symmetric manner**

**Name of Student:Aayush Chounkar**

**Roll Number: 04**

**ExperimentN0–16**

---

**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)**

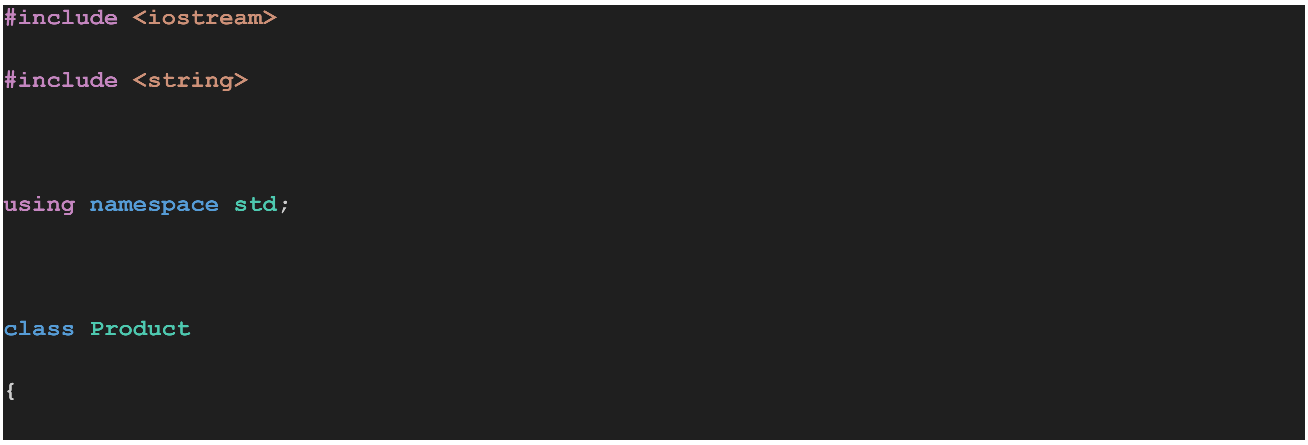
**• Price (a floating-point number)**

**• 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-Object-oriented principles in C++ are employed to design an inventory management system. • The Product class encapsulates attributes such as Product ID, Product Name, Price, and Quantity in Stock, with a parameterized constructor facilitating the initialization of these attributes for each new product**

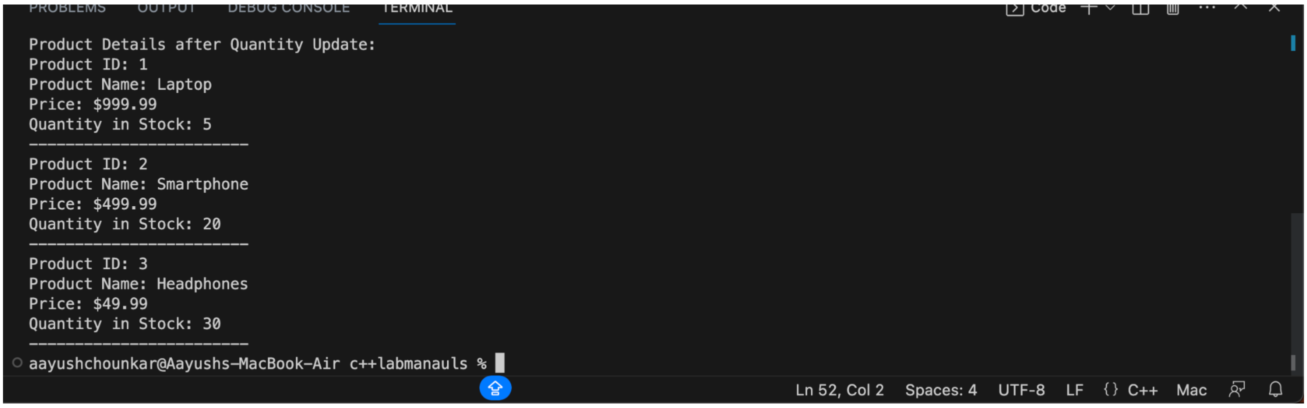
## CODE



```
#include <iostream>

#include <string>


using namespace std;


class Product

{
```

```cpp
private:
    int productId;
    string productName;
    float price;
    int quantityInStock;

public:
    Product(int id, const string &name, float p, int quantity)
        : productId(id), productName(name), price(p), quantityInStock(quantity) {}

    void displayProductDetails()
    {
        cout << "Product ID: " << productId << endl;
        cout << "Product Name: " << productName << endl;
        cout << "Price: $" << price << endl;
        cout << "Quantity in Stock: " << quantityInStock << endl;
        cout << "-------------------------" << endl;
    }

    void updateQuantity(int quantity)
    {
        quantityInStock += quantity;
    }
};

int main()
{
    Product product1(1, "Laptop", 999.99, 10);
    Product product2(2, "Smartphone", 499.99, 20);
    Product product3(3, "Headphones", 49.99, 30);
    cout << "Initial Product Details:" << endl;
    product1.displayProductDetails();
    product2.displayProductDetails();
    product3.displayProductDetails();

    product1.updateQuantity(-5);

    cout << "Product Details after Quantity Update:" << endl;
    product1.displayProductDetails();
    product2.displayProductDetails();
    product3.displayProductDetails();

    return 0;
}
```

## OUTPUT



```
Product Details after Quantity Update:
Product ID: 1
Product Name: Laptop
Price: $999.99
Quantity in Stock: 5
------------------------
Product ID: 2
Product Name: Smartphone
Price: $499.99
Quantity in Stock: 20
------------------------
Product ID: 3
Product Name: Headphones
Price: $49.99
Quantity in Stock: 30
------------------------
aayushchounkar@Aayushs-MacBook-Air c++labmanauls %
```

**CONCLUSION: • The program establishes a foundation for a small store's inventory management. • Utilizing the Product class allows for the creation, storage, and retrieval of product details, adhering to object-oriented principles for modularity and reusability in an inventory management system.**

**Name of Student:Aayush  Chounkar**

**Roll Number: 04**

**ExperimentN0–17**

---

**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

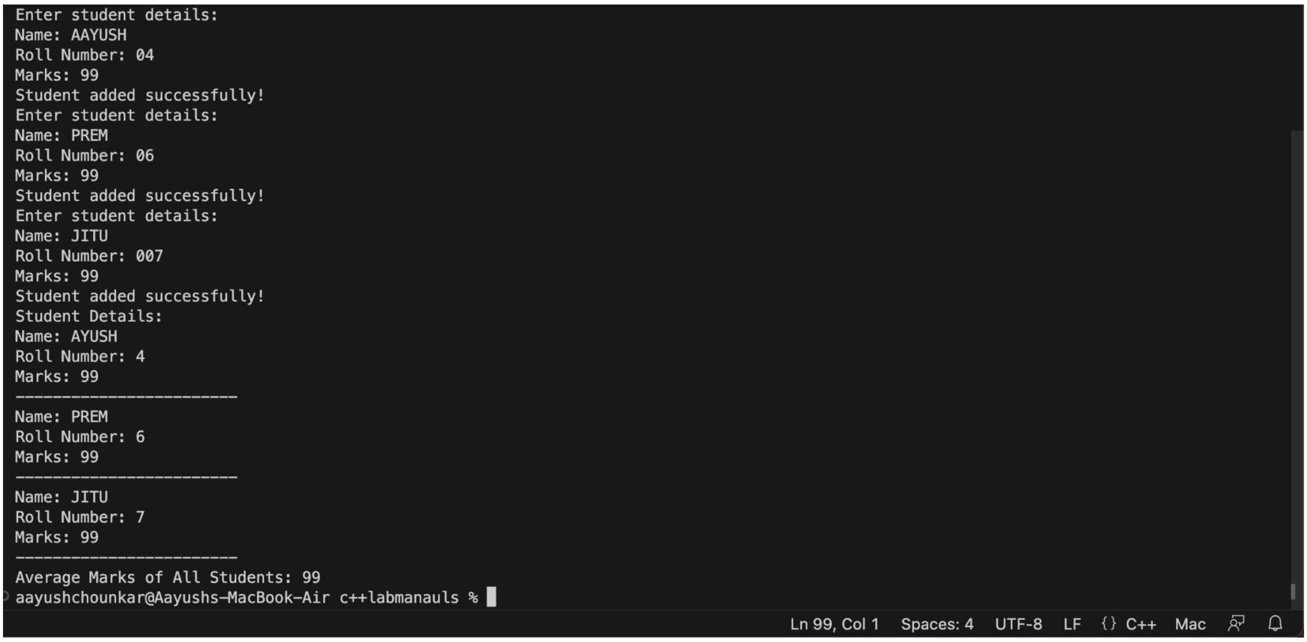
**The program utilizes object-oriented principles to manage student records in C++. • The Student class is designed with attributes such as name, roll number, and marks, and methods for displaying student details. Another class, StudentRecord, manages a collection of student objects, providing methods to add new students and calculate the average marks of all students in the record system.**

## CODE

```
#include <iostream>

using namespace std;


class Student {

private:

   string name;

   int rollNumber;

   float marks;
```

```
public:

    Student(const string& studentName, int studentRollNumber, float studentMarks)
        : name(studentName), rollNumber(studentRollNumber), marks(studentMarks) {}

    void displayStudentDetails() const {
        cout << "Name: " << name << endl;
        cout << "Roll Number: " << rollNumber << endl;
        cout << "Marks: " << marks << endl;
        cout << "------------------------\n";
    }

    const string& getName() const { return name; }
    void setName(const string& studentName) { name = studentName; }

    int getRollNumber() const { return rollNumber; }
    void setRollNumber(int studentRollNumber) { rollNumber = studentRollNumber; }

    float getMarks() const { return marks; }
    void setMarks(float studentMarks) { marks = studentMarks; }
};

class StudentRecordSystem {
private:
    vector<Student> studentRecords;

public:

    void addStudent() {
        string name;
        int rollNumber;
        float marks;

        cout << "Enter student details:\n";
        cout << "Name: ";
        cin.ignore();
        getline(cin, name);

        cout << "Roll Number: ";
        cin >> rollNumber;

        cout << "Marks: ";
        cin >> marks;

        studentRecords.push_back(Student(name, rollNumber, marks));
        cout << "Student added successfully!\n";
    }

    void displayAllStudents() const {
        if (studentRecords.empty()) {
            cout << "No student records available.\n";
        } else {
            cout << "Student Details:\n";
            for (const auto& student : studentRecords) {
                student.displayStudentDetails();
            }
        }
    }
    void calculateAndDisplayAverageMarks() const {
        if (studentRecords.empty()) {
            cout << "No student records available.\n";
        } else {
            float totalMarks = 0;
            for (const auto& student : studentRecords) {
                totalMarks += student.getMarks();
            }

            float averageMarks = totalMarks / studentRecords.size();
            cout << "Average Marks of All Students: " << averageMarks << std::endl;
        }
    }
};

int main() {

    StudentRecordSystem recordSystem;

    recordSystem.addStudent();
    recordSystem.addStudent();
    recordSystem.addStudent();

    recordSystem.displayAllStudents();
    recordSystem.calculateAndDisplayAverageMarks();

    return 0;
}
```

## OUTPUT



```
Enter student details:
Name: AAYUSH
Roll Number: 04
Marks: 99
Student added successfully!
Enter student details:
Name: PREM
Roll Number: 06
Marks: 99
Student added successfully!
Enter student details:
Name: JITU
Roll Number: 007
Marks: 99
Student added successfully!
Student Details:
Name: AYUSH
Roll Number: 4
Marks: 99
-------------------------
Name: PREM
Roll Number: 6
Marks: 99
-------------------------
Name: JITU
Roll Number: 7
Marks: 99
-------------------------
Average Marks of All Students: 99
aayushchounkar@Aayushs-MacBook-Air c++labmanauls %
```

**Conclusion:This basic student records management system provides a foundation for tracking and managing student information. Depending on your specific requirements, you can extend this program by adding more features such as updating student details, searching for a student by roll number, or storing data persistently in a database. This example serves as a starting point that can be customized and expanded based on the needs of the application.**

**Name of Student:Aayush  Chounkar**

**Roll Number: 04**

**ExperimentN0–18**

---

**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

*** The program implements a basic calculator using the C++ programming language.**

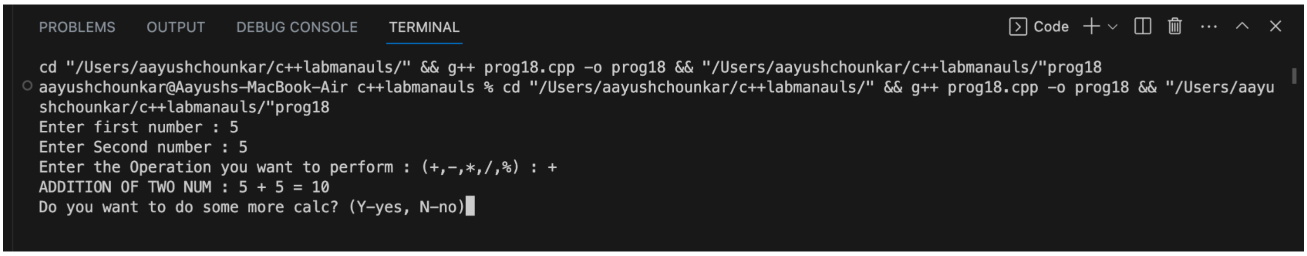
**• A class named Calculator is designed to perform four basic arithmetic operations: addition, subtraction, multiplication, and division.**

## CODE

```
#include <iostream>

using namespace std;


class Calc

{

private:

   int a, b;

   char opr;

   double sum;


public:

   void getnum()

   {
       cout << "Enter first number : ";

       cin >> a;


       cout << "Enter Second number : ";

       cin >> b;

   rerun:

       cout << "Enter the Operation you want to perform : (+,-,*,/,%) : ";

       cin >> opr;


       switch (opr)

       {

       case '+':

           sum = addition(a, b);

           cout << "ADDITION OF TWO NUM : " << a << " + " << b << " = " << sum;

           break;

       case '-':

           sum = subtraction(a, b);

           cout << "SUBTRACTION OF TWO NUM : " << a << " - " << b << " = " << sum;

           break;

       case '*':

           sum = multiplication(a, b);

           cout << "MULTIPLICATION OF TWO NUM : " << a << " x " << b << " = " << sum;

           break;

       case '/':

           sum = division(a, b);

           cout << "DIVISION OF TWO NUM : " << a << " / " << b << " = " << sum;

           break;

       case '%':

           sum = modulus(a, b);
```

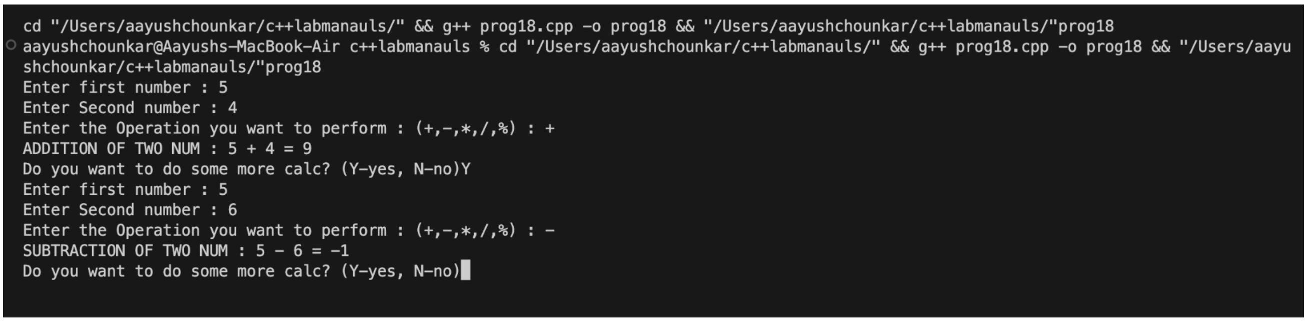
```
            cout << "REMAINDER OF TWO NUM DIVISON : " << a << " & " << b << " = " << modulus(a, b);

            break;

        default:

            cout << "WRONG INPUT \n\n";

            goto rerun;

        }

    }


    double addition(double a, double b)

    {

        return a + b;

    }


    double multiplication(double a, double b)

    {

        int c, d;

        double mult;


        c = a * 1000;

        d = b * 1000;


        mult = (c * d) / 1000000;

        return mult;

    }


    double division(double a, double b)

    {

        int c, d;

        double mult;
        c = a;

        d = b;


        mult = int(c / d);

        return mult;

    }


    double subtraction(double a, double b)

    {

        return a - b;

    }


    double modulus(double a, double b)

    {

        return int(a) % int(b);

    }


    void showhistory()

    {

        cout << a << " " << opr << " " << b << " = " << sum << endl;

    }

};


int main()

{

    int n = 100, i = 0;


    Calc cal[n];

    char x, y;
    do
    {

        cal[i].getnum();
        i++;
        cout << "\nDo you want to do some more calc? (Y-yes, N-no)";
        cin >> y;
        if (toupper(y) == 'N')
        {
            break;
        }
    } while (i < n);

    cout << "\n\nCalc HISTORY : \n";
    for (int j = 0; j < i; j++)
    {
        cal[j].showhistory();
    }

    return 0;
}
```

## OUTPUT



```
cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog18.cpp -o prog18 && "/Users/aayushchounkar/c++labmanauls/"prog18
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog18.cpp -o prog18 && "/Users/aayu
shchounkar/c++labmanauls/"prog18
Enter first number : 5
Enter Second number : 5
Enter the Operation you want to perform : (+,-,*,/,%) : +
ADDITION OF TWO NUM : 5 + 5 = 10
Do you want to do some more calc? (Y-yes, N-no)
```

## TESTCASE



```
cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog18.cpp -o prog18 && "/Users/aayushchounkar/c++labmanauls/"prog18
aayushchounkar@Aayushs-MacBook-Air c++labmanauls % cd "/Users/aayushchounkar/c++labmanauls/" && g++ prog18.cpp -o prog18 && "/Users/aayu
shchounkar/c++labmanauls/"prog18
Enter first number : 5
Enter Second number : 4
Enter the Operation you want to perform : (+,-,*,/,%) : +
ADDITION OF TWO NUM : 5 + 4 = 9
Do you want to do some more calc? (Y-yes, N-no)Y
Enter first number : 5
Enter Second number : 6
Enter the Operation you want to perform : (+,-,*,/,%) : -
SUBTRACTION OF TWO NUM : 5 - 6 = -1
Do you want to do some more calc? (Y-yes, N-no)
```

## CONCLUSION

- **The Calculator class provides methods for each arithmetic operation, enhancing modularity.**
- **The program allows the user to input two numbers and select an operation, ensuring basic calculator functionality with error handling for division by zero.**

**Name of Student:Aayush  Chounkar**

**Roll Number: 04**

**ExperimentN0–19**

---

**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-

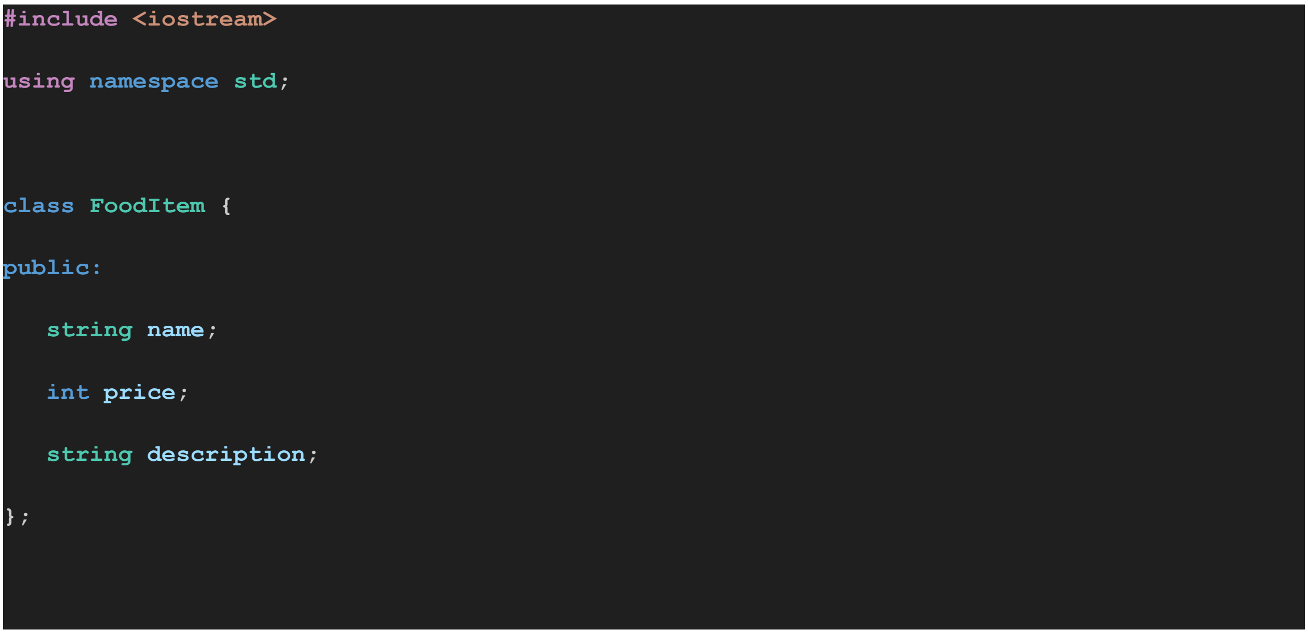
**• The program models a simple online food ordering system using classes and vectors.**

**• Utilizes object-oriented programming concepts, encapsulating items, menu, and orders.**

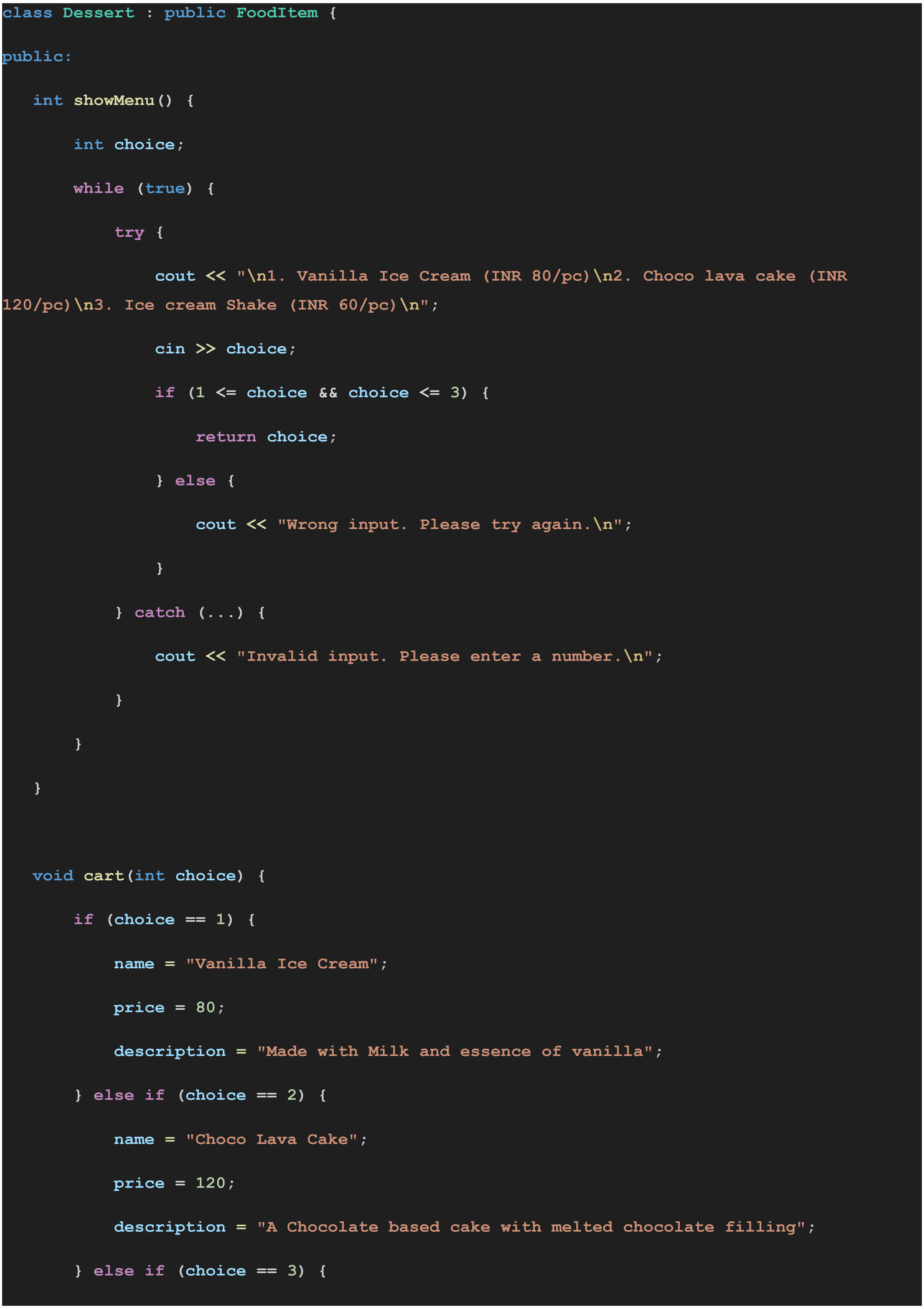
**• Demonstrates a menu-driven interface, allowing users to add items to the cart**

**. • Facilitates a seamless interaction for ordering food, enhancing user experience.**
**• Emphasizes modularity, scalability, and code readability in C++ programming.**

## CODE



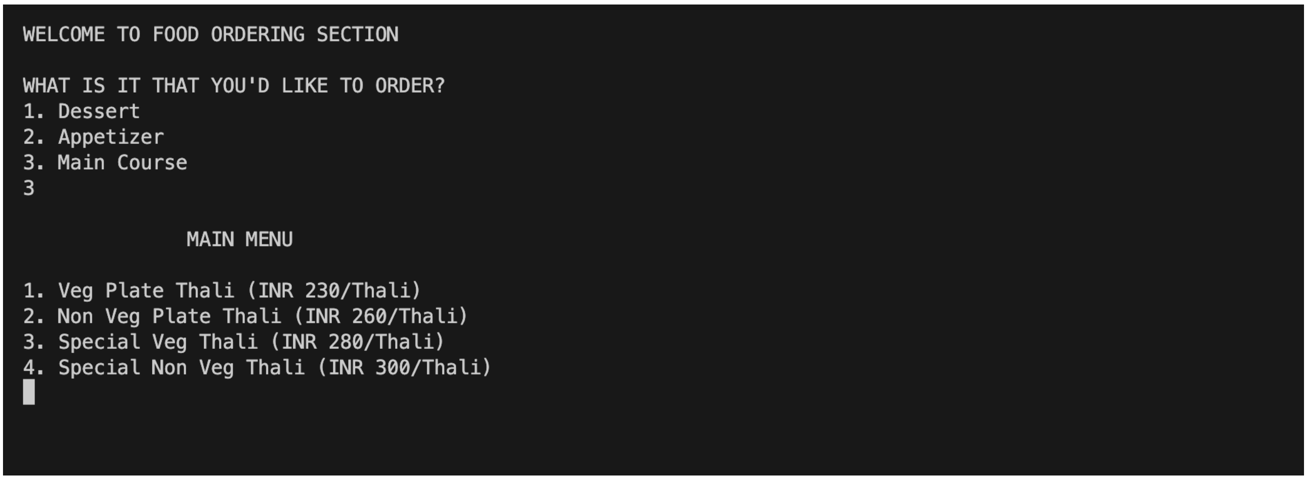
```
#include <iostream>

using namespace std;


class FoodItem {

public:

   string name;

   int price;

   string description;

};
```



```
class Dessert : public FoodItem {
public:
    int showMenu() {
        int choice;
        while (true) {
            try {
                cout << "\n1. Vanilla Ice Cream (INR 80/pc)\n2. Choco lava cake (INR
120/pc)\n3. Ice cream Shake (INR 60/pc)\n";
                cin >> choice;
                if (1 <= choice && choice <= 3) {
                    return choice;
                } else {
                    cout << "Wrong input. Please try again.\n";
                }
            } catch (...) {
                cout << "Invalid input. Please enter a number.\n";
            }
        }
    }

    void cart(int choice) {
        if (choice == 1) {
            name = "Vanilla Ice Cream";
            price = 80;
            description = "Made with Milk and essence of vanilla";
        } else if (choice == 2) {
            name = "Choco Lava Cake";
            price = 120;
            description = "A Chocolate based cake with melted chocolate filling";
        } else if (choice == 3) {
            name = "Ice Cream Shake";

            price = 60;

            description = "Milk Shake with ice cream topped on it, a mouthwatering edible
 shake";

        }

    }

 };


 class Appetizer : public FoodItem {

 public:

    int showMenu() {

        int choice;

        while (true) {

            try {

                cout << "\n1. Potato Wedges (INR 120/plate)\n2. French Fries (INR
 100/plate)\n3. Paneer Chilly (INR 180/plate)\n";

                cin >> choice;

                if (1 <= choice && choice <= 3) {

                    return choice;

                } else {

                    cout << "Wrong input. Please try again.\n";

                }

            } catch (...) {

                cout << "Invalid input. Please enter a number.\n";

            }

        }

    }

        void cart(int choice) {

        if (choice == 1) {

            name = "Potato Wedges";

            price = 120;
```

```
            description = "Fried Potato with wrinkled shape and seasoned with salt and pepper";
        } else if (choice == 2) {
            name = "French Fries";
            price = 100;
            description = "Deep Fried Potato sticks with Peri Peri seasonings";
        } else if (choice == 3) {
            name = "Paneer Chilly";
            price = 180;
            description = "Paneer Seasoned with veggies and sauce and Saute with golden crisp";
        }
    }
};

class MainCourse : public FoodItem {
public:
    int showMenu() {
        int choice;
        while (true) {
            try {
                cout << "\n1. Veg Plate Thali (INR 230/Thali)\n2. Non Veg Plate Thali (INR 260/Thali)\n3. Special Veg Thali (INR 280/Thali)\n4. Special Non Veg Thali (INR 300/Thali)\n";
                cin >> choice;
                if (1 <= choice && choice <= 4) {
                    return choice;
                } else {
                    cout << "Wrong input. Please try again.\n";
                }
            } catch (...) {
                cout << "Invalid input. Please enter a number.\n";
            }
        }
    }

    void cart(int choice) {
        if (choice == 1) {
            name = "Veg Plate Thali";
            price = 230;
            description = "A Full meal Thali Served with 4 Rotis and Jeera Rice alongside With salad, paneer, Dal, papad";
        } else if (choice == 2) {
            name = "Non Veg Plate Thali";
            price = 260;
            description = "A Full meal Thali Served with 4 Rotis and Jeera Rice alongside With salad, Chicken, Dal, papad";
        } else if (choice == 3) {
            name = "Special Veg Thali";
            price = 280;
            description = "A Full meal Thali Served with 4 Rotis and Jeera Rice alongside With salad, 2 paneer sabzi, veg mix, Dal, papad";
        } else if (choice == 4) {
            name = "Special Non Veg Thali";
            price = 300;
            description = "A Full meal Thali Served with 4 Rotis and Jeera Rice alongside With salad, 2 Chicken sabzis, Mutton, Dal, papad";
        }
    }
};

void foodMenu() {
    int price = 0;

    string pay, id, y;

    Dessert des;

    Appetizer app;

    MainCourse mncr;

    int i = 1;

    vector<string> l;

    while (true) {

        cout << "\nWELCOME TO FOOD ORDERING SECTION\n\nWHAT IS IT THAT YOU'D LIKE TO ORDER?";

        cout << "\n1. Dessert\n2. Appetizer\n3. Main Course\n";

        int choice;

        cin >> choice;


        if (choice > 3 || choice < 1) {

            continue;

        }

        if (choice == 1) {

            cout << "\n               DESSERT                  \n";

            int dc = des.showMenu();

            des.cart(dc);

        } else if (choice == 2) {

            cout << "\n               APPETIZER                  \n";

            int ac = app.showMenu();

            app.cart(ac);

        } else if (choice == 3) {

            cout << "\n              MAIN MENU                  \n";

            int mc = mncr.showMenu();

            mncr.cart(mc);

        }
        if (choice == 1) {
            cout << "\n\nYour Item : ";
            cout << "\n           " << des.name << "  for INR " << des.price << "\n
" << des.description << endl;
            l.push_back(des.name + "  for INR " + to_string(des.price) + "           " +
des.description);
            price += des.price;
        } else if (choice == 2) {
            cout << "Your Item : ";
            cout << "\n           " << app.name << "  for INR " << app.price << "\n
" << app.description << endl;
            l.push_back(app.name + "  for INR " + to_string(app.price) + "            " +
app.description);
            price += app.price;
        } else if (choice == 3) {
            cout << "Your Item : \n\n           " << mncr.name << "  for INR " <<
mncr.price << "\n           " << mncr.description << endl;
            l.push_back(mncr.name + "  for INR " + to_string(mncr.price) + "            " +
mncr.description);
            price += mncr.price;
        }


        cout << "\n\nWant to order any other food? (y/n)";
        cin >> y;


        if (y[0] != 'Y' && y[0] != 'y') {
            cout << "\n\nEAT WELL :)\nYOUR BILL : \n\n\n";
            int a = 1;
            for (const auto &item : l) {
                cout << a << " : " << item << endl;
                a++;
            }


            cout << "\nTotal Bill = INR " << price + (price * 0.18) << " + (TAX)  " << price * 0.18 << endl;
            cout << "\nGRAND TOTAL  = INR " << price + (price * 0.18) << endl;
            cout << "\nHow would you like to pay? (Online/Cash)";
            cin >> pay;
            if (pay == "ONLINE" || pay == "CASH") {
                if (pay == "ONLINE") {
                    cout << "\nEnter Upi id : ";
                    cin >> id;
                    cout << "\nSent the request to pay " << price + (price * 0.18) << " to your upi id " << id << endl;
                } else {
                    cout << "\nPlease pay " << price + (price * 0.18) << " with the cash With Change......Thank you :)" << endl;
                }
            }
            cout << "\nTHANK YOU FOR CHOOSING OUR TREAT. HAVE A GREAT DINE :)" << endl;
            break;
        }
    }
}


int main() {
    foodMenu();
    return 0;
}
```

## OUTPUT



```
WELCOME TO FOOD ORDERING SECTION

WHAT IS IT THAT YOU'D LIKE TO ORDER?
1. Dessert
2. Appetizer
3. Main Course
3

              MAIN MENU

1. Veg Plate Thali (INR 230/Thali)
2. Non Veg Plate Thali (INR 260/Thali)
3. Special Veg Thali (INR 280/Thali)
4. Special Non Veg Thali (INR 300/Thali)
```

## CONCLUSION

**CONCLUSION: • Simulates an online food ordering system with a menu and cart. • Allows users to add items to their cart, displaying details and total cost. • Provides a user-friendly experience for ordering food online.**

**Name of Student:Aayush  Chounkar**

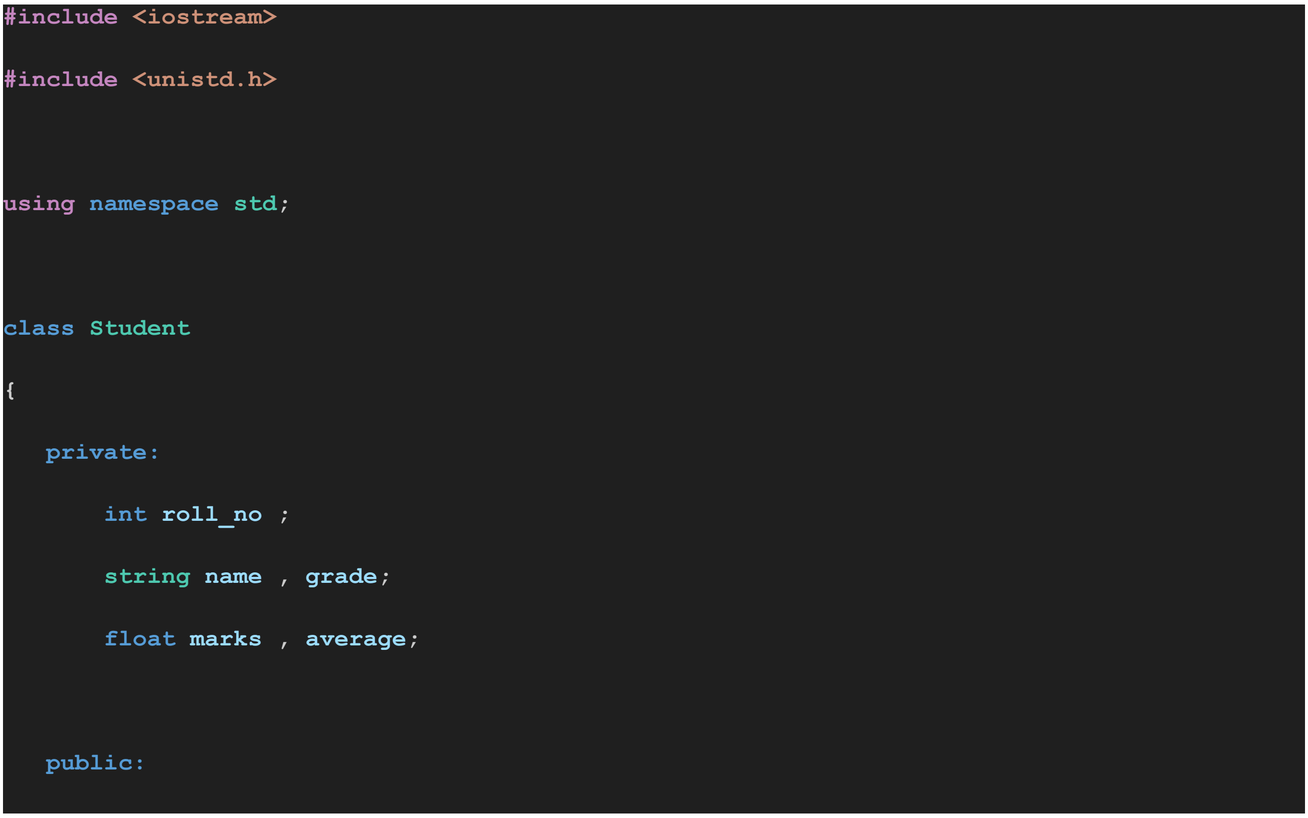
**Roll Number: 04**

**ExperimentN0–20**

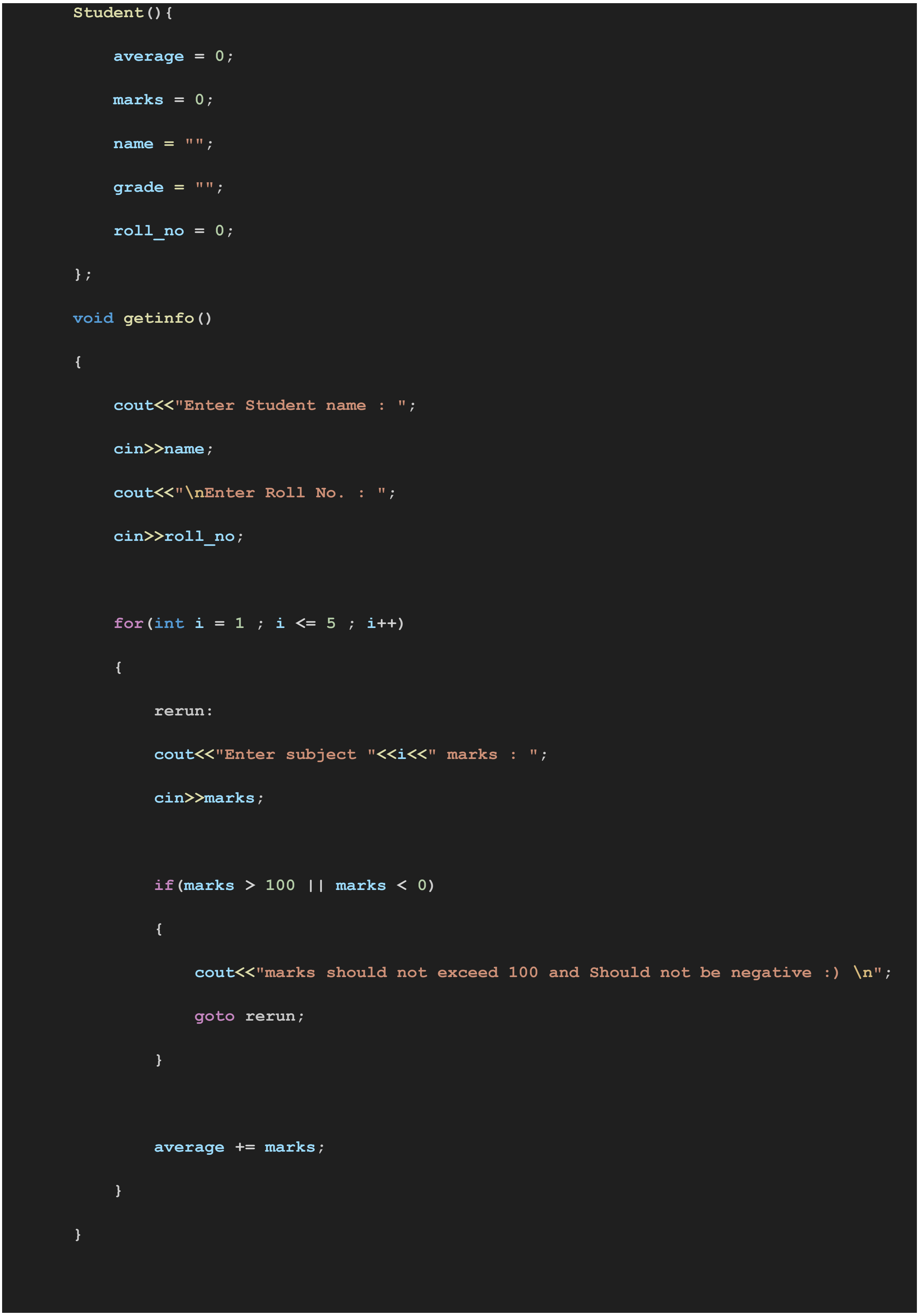
**TITLE: Write a program to manage student grades for a classroom. Create a class Student with attributes for student name and an array to store grades. Implement methods for adding grades, calculating the average grade, and displaying the student's name and grades. Use constructors and destructors to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

**THEORY: • Student class manages grades with encapsulation. • Constructors initialize, and destructors release resources.**

**CODE-**

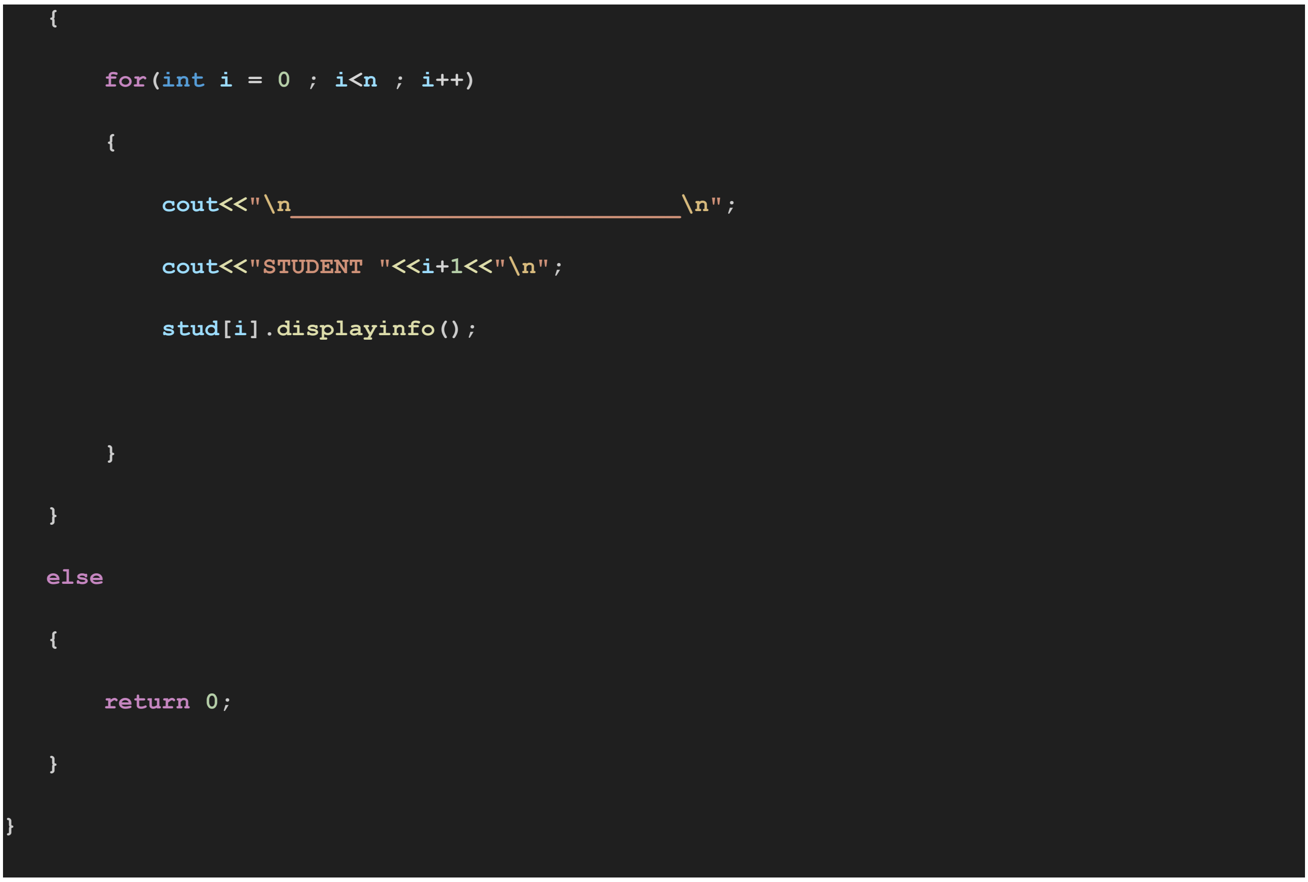


```
#include <iostream>

#include <unistd.h>


using namespace std;


class Student

{

   private:

       int roll_no ;

       string name , grade;

       float marks , average;


   public:
```



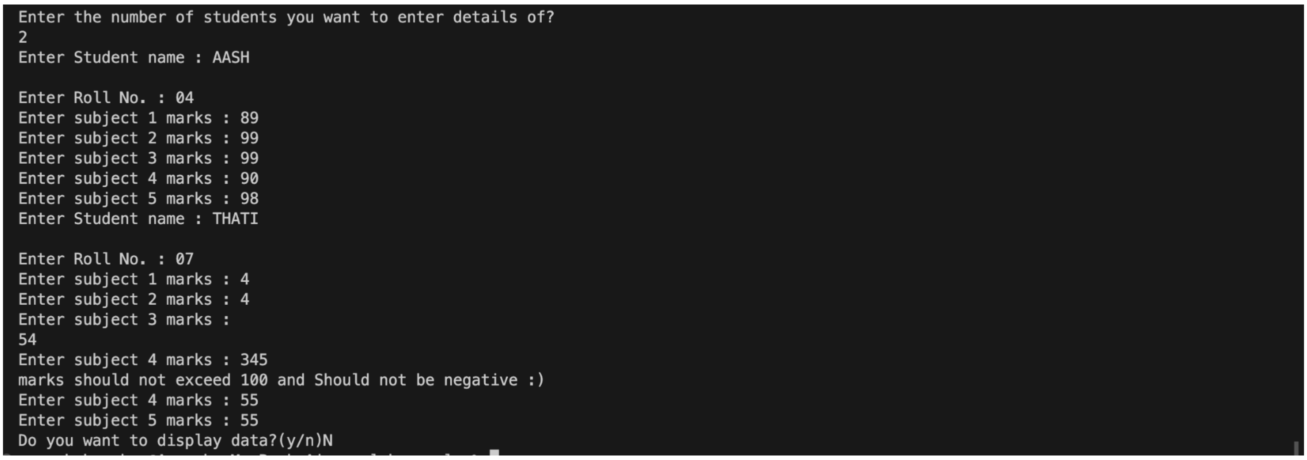
```
    Student(){
        average = 0;
        marks = 0;
        name = "";
        grade = "";
        roll_no = 0;
    };
    void getinfo()
    {
        cout<<"Enter Student name : ";
        cin>>name;
        cout<<"\nEnter Roll No. : ";
        cin>>roll_no;

        for(int i = 1 ; i <= 5 ; i++)
        {
            rerun:
            cout<<"Enter subject "<<i<<" marks : ";
            cin>>marks;

            if(marks > 100 || marks < 0)
            {
                cout<<"marks should not exceed 100 and Should not be negative :) \n";
                goto rerun;
            }

            average += marks;
        }
    }
    void displayinfo()
    {
        cout<<"NAME : "<<name<<"\n";
        cout<<"ROLL NO. : "<<roll_no<<"\n";
        cout<<"Total marks(out of 500) : "<<average<<"\n";

        average /= 5.00;
        if(average >= 85 && average < 95)
        {
            cout<<"PERCENTAGE : "<<average<<"% with GRADE : A";
        }
        else if(average >= 95)
        {
            cout<<"PERCENTAGE : "<<average<<"% with GRADE : A+";
        }
        else if(average >= 75 && average < 85)
        {
            cout<<"PERCENTAGE : "<<average<<"% with GRADE : B";
        }
        else if(average < 75 && average >= 60)
        {
            cout<<"PERCENTAGE : "<<average<<"% with GRADE : C";
        }
        else if(average < 60 && average > 33)
        {
            cout<<"PERCENTAGE : "<<average<<"% with GRADE : D";
        }
        else
        {
            cout<<"PERCENTAGE : "<<average<<"% and failed class with GRADE : F";
```

```
           }

           sleep(2);

       }


       ~Student(){


       }

};



int main()

{

   int n;

   char y;


   cout<<"\nEnter the number of students you want to enter details of? \n";

   cin>>n;


   Student stud[n];


   for(int i = 0 ; i<n ; i++)

   {

       stud[i].getinfo();

   }



   cout<<"Do you want to display data?(y/n)";

   cin>>y;


   if(toupper(y) == 'Y')
```



```
    {

        for(int i = 0 ; i<n ; i++)

        {

            cout<<"\n___________________________\n";

            cout<<"STUDENT "<<i+1<<"\n";

            stud[i].displayinfo();


        }

    }

    else

    {

        return 0;

    }

}
```

# OUTPUT



```
Enter the number of students you want to enter details of?
2
Enter Student name : AASH

Enter Roll No. : 04
Enter subject 1 marks : 89
Enter subject 2 marks : 99
Enter subject 3 marks : 99
Enter subject 4 marks : 90
Enter subject 5 marks : 98
Enter Student name : THATI

Enter Roll No. : 07
Enter subject 1 marks : 4
Enter subject 2 marks : 4
Enter subject 3 marks :
54
Enter subject 4 marks : 345
marks should not exceed 100 and Should not be negative :)
Enter subject 4 marks : 55
Enter subject 5 marks : 55
Do you want to display data?(y/n)N
```

## CONCLUSION

- **Effective encapsulation ensures organized grade management.**
- **Constructors and destructors enhance resource handling and code readability.**