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

C++ PROGRAMMING LAB

Prepared by:

Name of Student: Lakshya Duhoon

Roll No: 46

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: **Shah Mohd Areeb Mohd Aslam**

Roll Number: **21**

Experiment No: 1

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

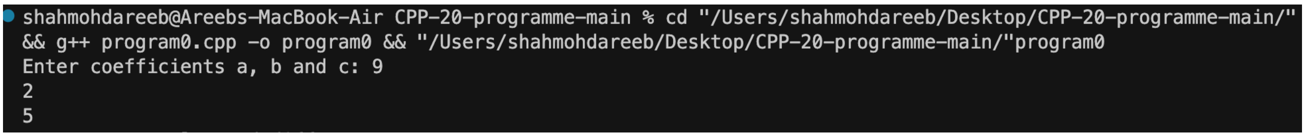
Theory:

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:

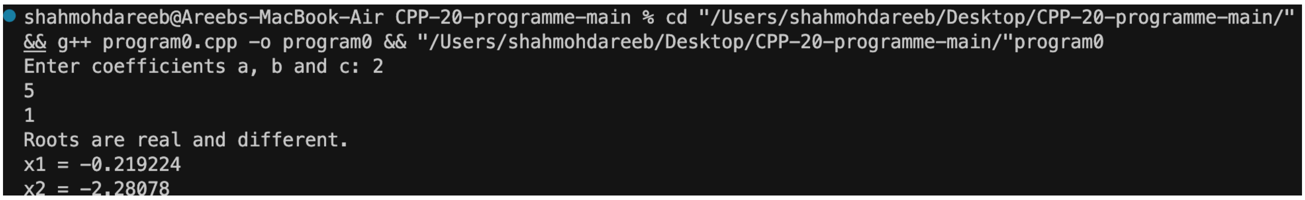
```
// to find the roots of a quadratic equation
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
    float a, b, c, x1, x2, discriminant, realPart, imaginaryPart;
    cout << "Enter coefficients a, b and c: ";
    cin >> a >> b >> c;
    // finding value of discriminant using values given by user
    discriminant = b * b - 4 * a * c;
    // if discriminant is postive, then roots are real and positive
by solving the quadratic equation
    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;
    }
    // if discriminant is 0, print the following message
    else if (discriminant == 0)
    {
        cout << "Roots are real and same." << endl;
        x1 = -b / (2 * a);
        cout << "x1 = x2 =" << x1 << endl;
    }
    // if discriminant is negative, then show real and imaginary
part
    else
    {
        realPart = -b / (2 * a);
        imaginaryPart = sqrt(-discriminant) / (2 * a);
        cout << "Roots are complex and different." << endl;
    }
    return 0;
}
```

Output: (screenshot)

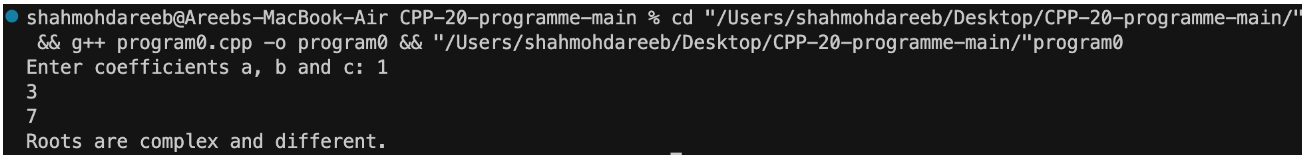


```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
&& g++ program0.cpp -o program0 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program0
Enter coefficients a, b and c: 9
2
5
```

Test Case: Any two (screenshot)



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
&& g++ program0.cpp -o program0 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program0
Enter coefficients a, b and c: 2
5
1
Roots are real and different.
x1 = -0.219224
x2 = -2.28078
```



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
 && g++ program0.cpp -o program0 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program0
Enter coefficients a, b and c: 1
3
7
Roots are complex and different.
```

Conclusion:

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

Name of Student: Shah Mohd Areeb Mohd Aslam

Roll Number: 21

Experiment No: 2

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

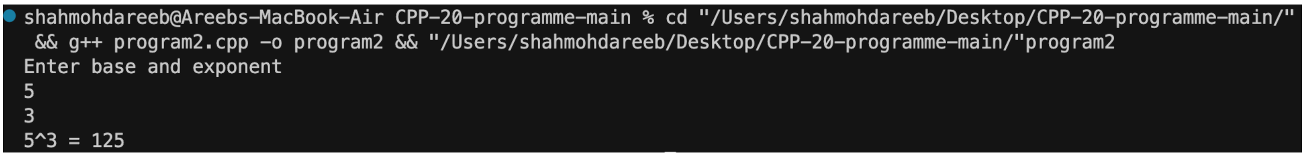
Theory:

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:

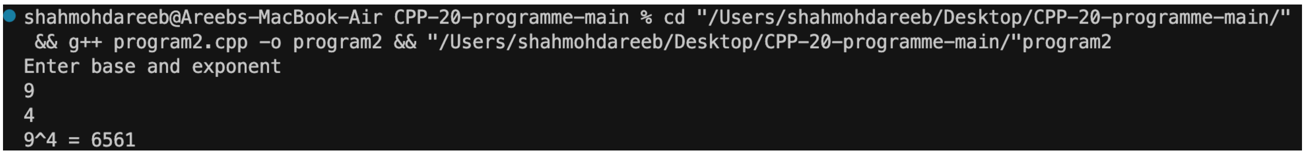
```
//program to calculate power of a number using loop
#include <iostream>

using namespace std;

int main(){
    int base, exp, i, result = 1;

 cout << "Enter base and exponent\n";
    cin >> base >> exp;

    // Calculate base^exponent by repetitively multiplying base
    for(i = 0; i < exp; i++){
        result = result * base;
    }

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

    return 0;
}
```

Output: (screenshot)

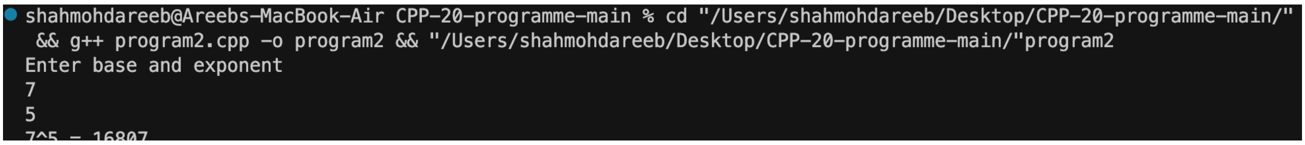


```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
 && g++ program2.cpp -o program2 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program2
Enter base and exponent
5
3
5^3 = 125
```

Test Case: Any two (screenshot)



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
 && g++ program2.cpp -o program2 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program2
Enter base and exponent
9
4
9^4 = 6561
```



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
 && g++ program2.cpp -o program2 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program2
Enter base and exponent
7
5
7^5 = 16807
```

Conclusion:

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

Name of Student: Shah Mohd Areeb Mohd Aslam

Roll Number: 21

Experiment No: 3

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

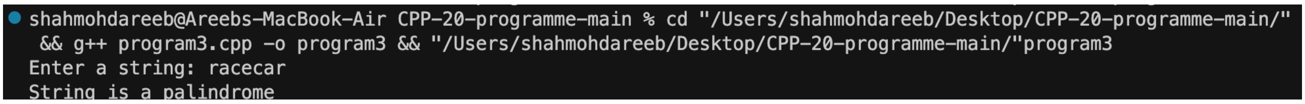
Theory:

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

Code:

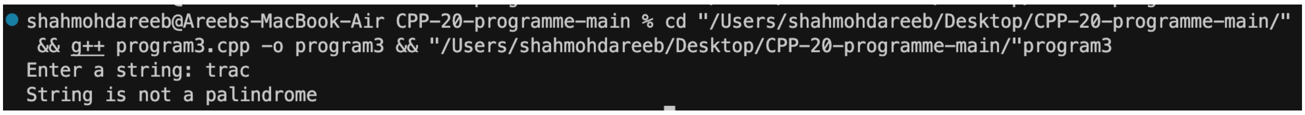
```
//program to check whether a given string is a palindrome
#include<iostream>
#include<string>
using namespace std;
int main()
{
    string str,str1;
    cout<<"Enter a string: ";
    cin>>str;
    str1=str;
    int start=0,end=str.length()-1;
    while(start<end)
    {
        char temp=str[start];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

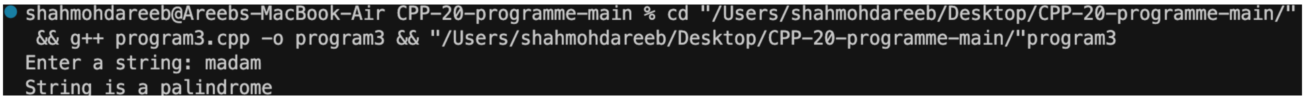


```
● shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
   && g++ program3.cpp -o program3 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program3
  Enter a string: racecar
  String is a palindrome
```

Test Case: Any two (screenshot)



```
● shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
   && g++ program3.cpp -o program3 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program3
  Enter a string: trac
  String is not a palindrome
```



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
 && g++ program3.cpp -o program3 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program3
Enter a string: madam
String is 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: Shah Mohd Areeb Mohd Aslam

Roll Number: 21

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:

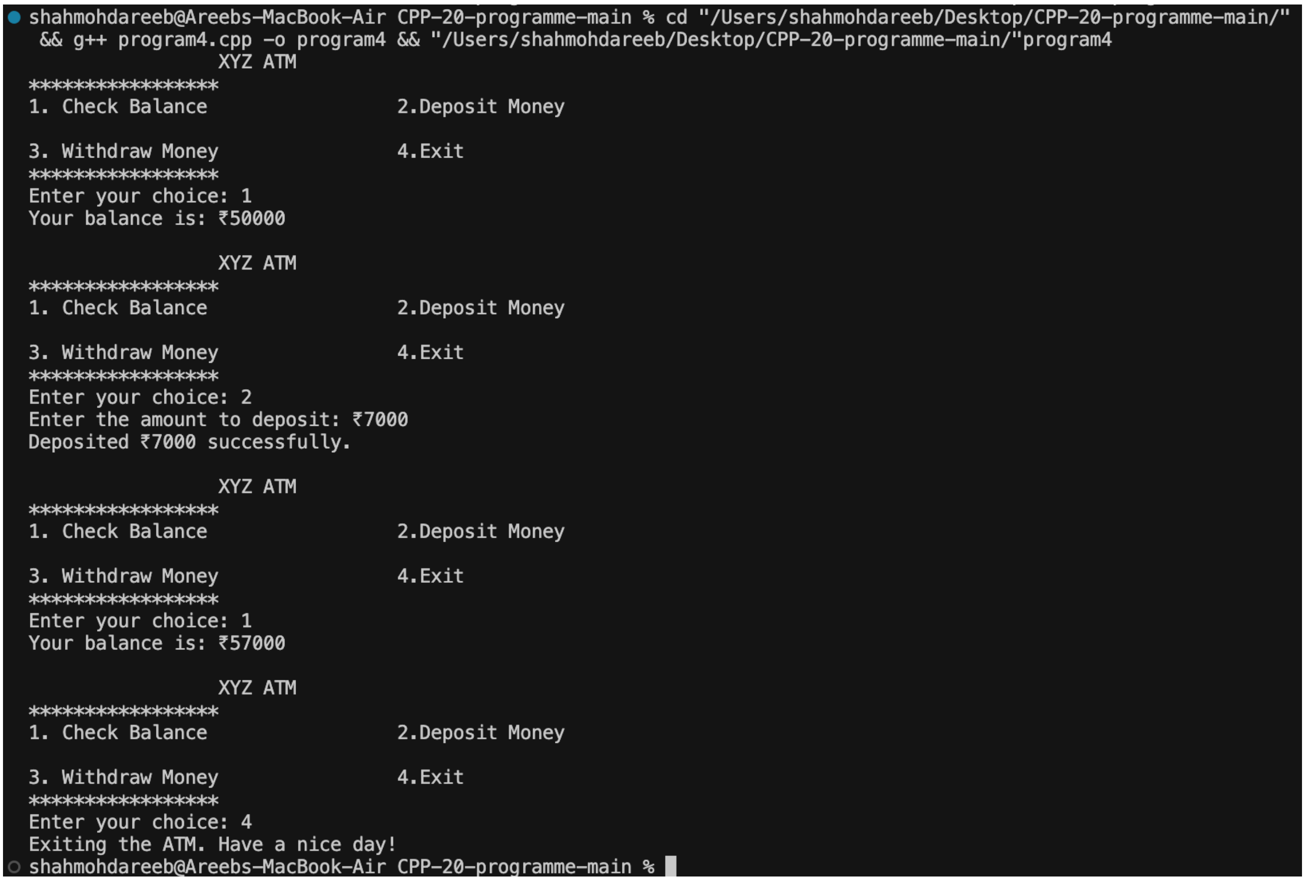
Using while loop and switch statement to print an ATM menu and take user’s choice and perform the respective operation.

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>
```

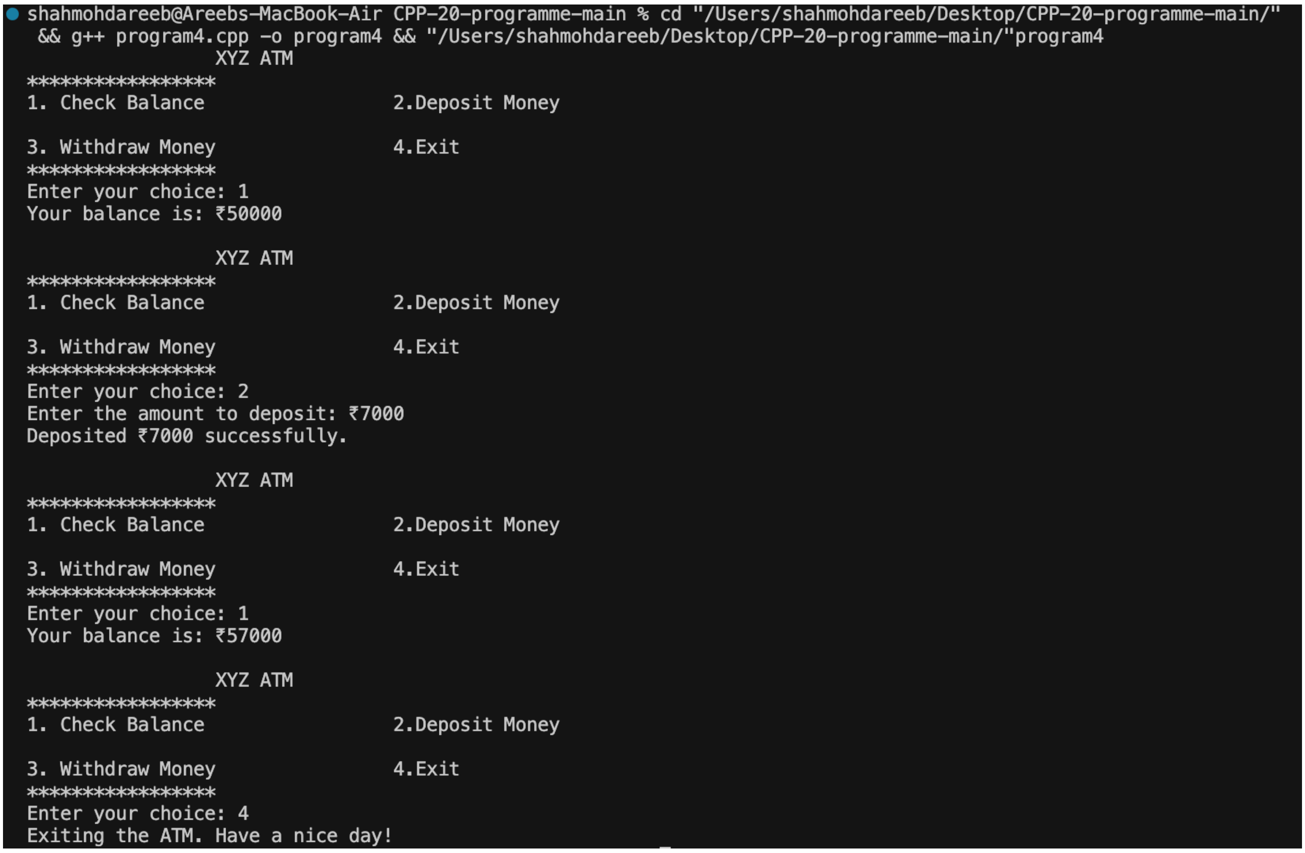
```
using namespace std;
int main() {
    //balance is 50000 Rs
    float balance = 50000.0;
    int choice;
    float amount;
    //to print infinitely
    while (true) {
        cout << "\t\t XYZ ATM \t\t" << endl;
        cout << "*****************" << endl;
        cout << "1. Check Balance \t\t 2.Deposit Money\n" << endl;
        cout << "3. Withdraw Money \t\t 4.Exit" << endl;
        cout << "*****************" << endl;
        cout << "Enter your choice: ";
        cin >> choice;
        //using switch loop to print a menu and let user choose 
which block of code they want to access
        //choice 1 to see account balance
        switch (choice) {
            case 1:
                cout << "Your balance is: ₹" << balance << 
"\n"<<endl;
                break;
            //choice 2 to deposit money (only if amount to deposit 
is >0)
            case 2:
                cout << "Enter the amount to deposit: ₹";
                cin >> amount;
                if (amount > 0) {
                    balance += amount;
                    cout << "Deposited ₹" << amount << " 
successfully.\n" << endl;
                } else {
                    cout << "Invalid amount. Please enter a 
positive amount.\n" << endl;
                }
                break;
            //choice 3 to withdraw money
            case 3:
                cout << "Enter the amount to withdraw: ₹";
                cin >> amount;
                //only allowed to withdraw if amount to be 
withdrawn <= account balance and is a positive number
                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;
            //choice 4 to exit the ATM
            case 4:
                cout << "Exiting the ATM. Have a nice day!" <<
endl;
                return 0;//ends entire main function's execution
            //default message to print if user enters a value other
than 1,2,3,4
            default:
                cout << "Invalid choice. Please select a valid 
option.\n" << endl;
                break;
        }
    }
    return 0;
}
```

Output: (screenshot)

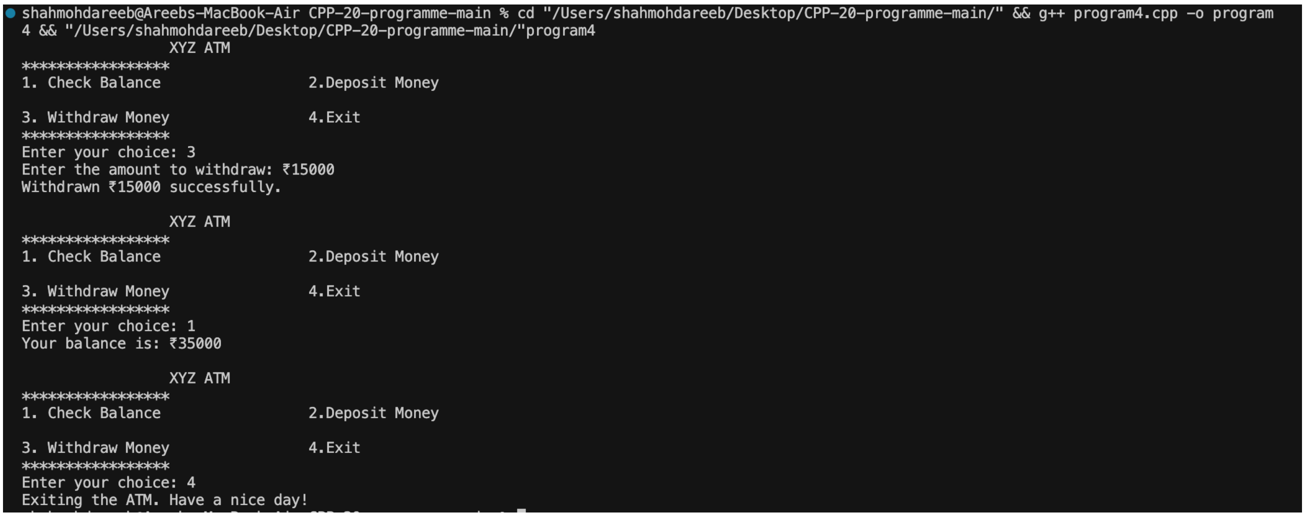


```
● shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
   && g++ program4.cpp -o program4 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program4
                   XYZ ATM
  *****************
  1. Check Balance                 2.Deposit Money

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

                   XYZ ATM
  *****************
  1. Check Balance                 2.Deposit Money

  3. Withdraw Money                4.Exit
  *****************
  Enter your choice: 2
  Enter the amount to deposit: ₹7000
  Deposited ₹7000 successfully.

                   XYZ ATM
  *****************
  1. Check Balance                 2.Deposit Money

  3. Withdraw Money                4.Exit
  *****************
  Enter your choice: 1
  Your balance is: ₹57000

                   XYZ ATM
  *****************
  1. Check Balance                 2.Deposit Money

  3. Withdraw Money                4.Exit
  *****************
  Enter your choice: 4
  Exiting the ATM. Have a nice day!
○ shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % █
```

Test Case: Any two (screenshot)



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
 && g++ program4.cpp -o program4 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program4
                 XYZ ATM
*****************
1. Check Balance                 2.Deposit Money

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

                 XYZ ATM
*****************
1. Check Balance                 2.Deposit Money

3. Withdraw Money                4.Exit
*****************
Enter your choice: 2
Enter the amount to deposit: ₹7000
Deposited ₹7000 successfully.

                 XYZ ATM
*****************
1. Check Balance                 2.Deposit Money

3. Withdraw Money                4.Exit
*****************
Enter your choice: 1
Your balance is: ₹57000

                 XYZ ATM
*****************
1. Check Balance                 2.Deposit Money

3. Withdraw Money                4.Exit
*****************
Enter your choice: 4
Exiting the ATM. Have a nice day!
```



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/" && g++ program4.cpp -o program
4 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program4
                 XYZ ATM
*****************
1. Check Balance                 2.Deposit Money

3. Withdraw Money                4.Exit
*****************
Enter your choice: 3
Enter the amount to withdraw: ₹15000
Withdrawn ₹15000 successfully.

                 XYZ ATM
*****************
1. Check Balance                 2.Deposit Money

3. Withdraw Money                4.Exit
*****************
Enter your choice: 1
Your balance is: ₹35000

                 XYZ ATM
*****************
1. Check Balance                 2.Deposit Money

3. Withdraw Money                4.Exit
*****************
Enter your choice: 4
Exiting the ATM. Have a nice day!
```

Conclusion:

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

Name of Student: Shah Mohd Areeb Mohd Aslam

Roll Number: 21

Experiment No: 5

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

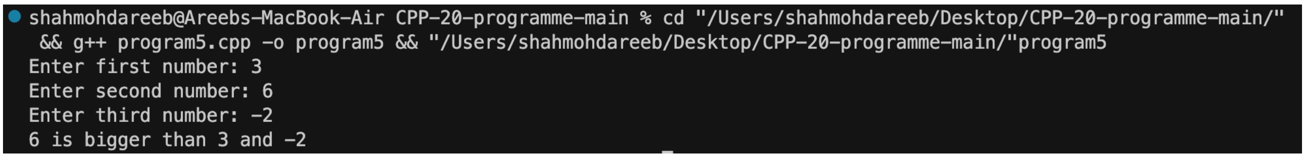
Theory:

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:

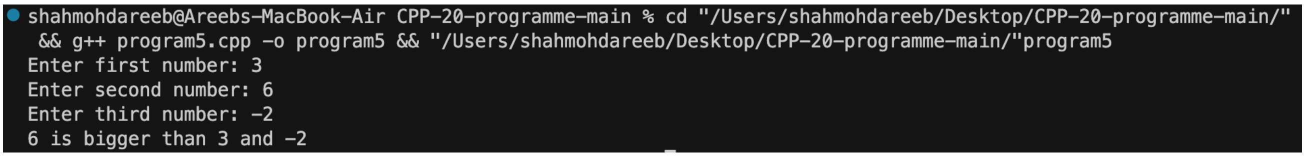
```cpp
// to find largest among three numbers using nested if else loop
#include <iostream>
using namespace std;
// main function
int main() {
    // input numbers from user
    int num1, num2, num3;
    cout << "Enter first number: ";
    cin >> num1;

    cout << "Enter second number: ";
    cin >> num2;
    cout << "Enter third number: ";
    cin >> num3;
    // check weather first number or second number or third number
is bagger among them
    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)

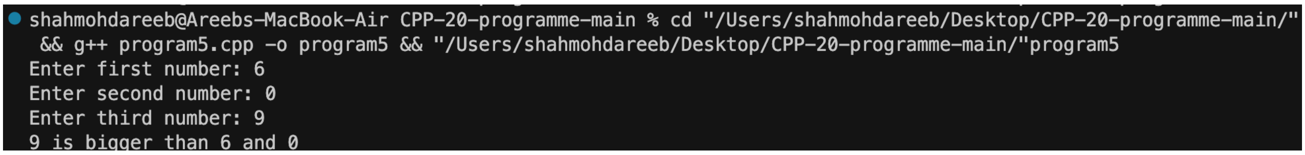


```
● shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
  && g++ program5.cpp -o program5 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program5
  Enter first number: 3
  Enter second number: 6
  Enter third number: -2
  6 is bigger than 3 and -2
```

Test Case: Any two (screenshot)



```
● shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
  && g++ program5.cpp -o program5 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program5
  Enter first number: 3
  Enter second number: 6
  Enter third number: -2
  6 is bigger than 3 and -2
```



```
● shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
  && g++ program5.cpp -o program5 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program5
  Enter first number: 6
  Enter second number: 0
  Enter third number: 9
  9 is bigger than 6 and 0
```

Conclusion:

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

Name of Student: Shah Mohd Areeb Mohd Aslam

Roll Number: 21

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.

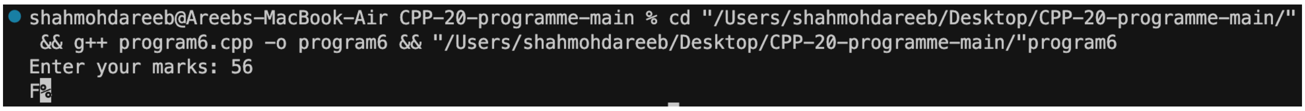
Theory:

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:

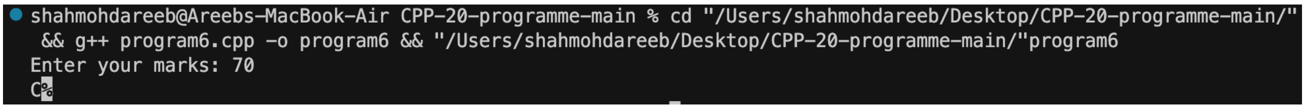
```
//Implement a program that determines the grade of a student based on their marks.
#include <iostream>
using namespace std;
int main() {
    float marks;
    cout << "Enter your marks: ";
    cin >> marks; //prints grade depending on the marks given using if else if loop
    if (marks >= 90) {
 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

Output: (screenshot)

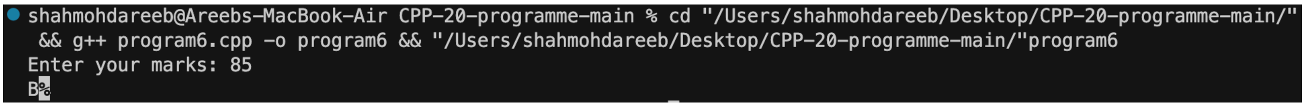


```
● shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
  && g++ program6.cpp -o program6 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program6
  Enter your marks: 56
  F%
```

Test Case: Any two (screenshot)



```
● shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
  && g++ program6.cpp -o program6 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program6
  Enter your marks: 70
  C%
```



```
● shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
  && g++ program6.cpp -o program6 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program6
  Enter your marks: 85
  B%
```

Conclusion:

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

Name of Student: Shah Mohd Areeb Mohd Aslam

Roll Number: 21

Experiment No: 7

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

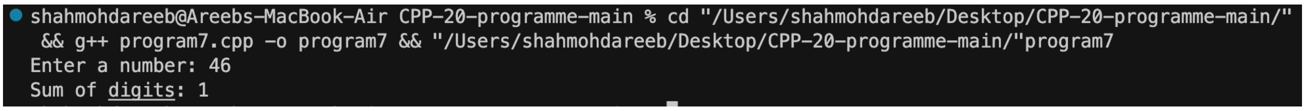
Theory:

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:

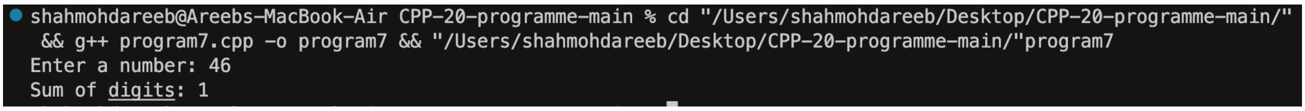
```
//program to find the sum of digits of a number until it becomes a 
single-digit number
#include<iostream>
using namespace std;
int main()
{
    int n,r;
    cout<<"Enter a number: ";
    cin>>n;
    while (n>=10)
    {
        int sum=0;
        while(n>0)
        {
            r=n%10;
        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

Output: (screenshot)

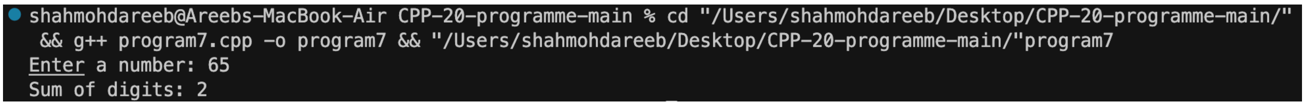


```
● shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
   && g++ program7.cpp -o program7 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program7
  Enter a number: 46
  Sum of digits: 1
```

Test Case: Any two (screenshot)



```
● shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
   && g++ program7.cpp -o program7 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program7
  Enter a number: 46
  Sum of digits: 1
```



```
● shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
   && g++ program7.cpp -o program7 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program7
  Enter a number: 65
  Sum of digits: 2
```

Conclusion:

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

Name of Student: Shah Mohd Areeb Mohd Aslam

Roll Number: 21

Experiment No: 8

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

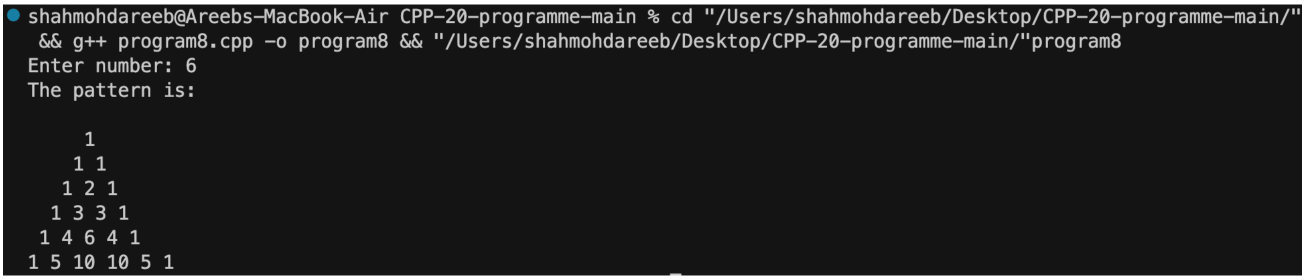
Theory:

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:

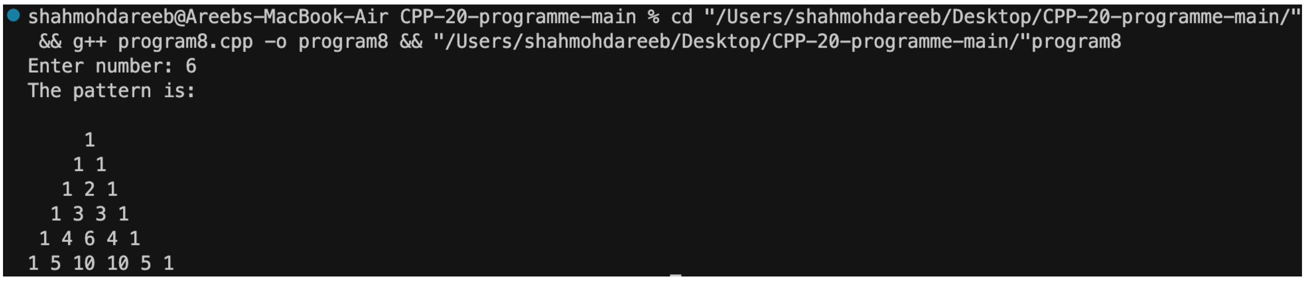
```
// pascal triangle
#include <iostream>
using namespace std;

// main function
int main()
{

    // input a number from user
    int n;
    cout << "Enter number: ";
    cin >> n;

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

    // displaying the pattern
    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)

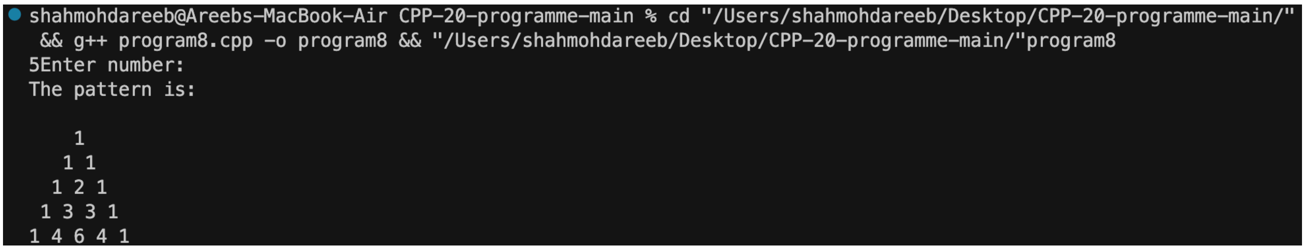


```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
 && g++ program8.cpp -o program8 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program8
Enter number: 6
The pattern is:

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

Test Case: Any two (screenshot)



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
 && g++ program8.cpp -o program8 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program8
Enter number: 6
The pattern is:

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



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
 && g++ program8.cpp -o program8 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program8
5Enter number:
The pattern is:

    1
   1 1
  1 2 1
 1 3 3 1
1 4 6 4 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: **Shah Mohd Areeb Mohd Aslam**

Roll Number: **21**

Experiment No: 9

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

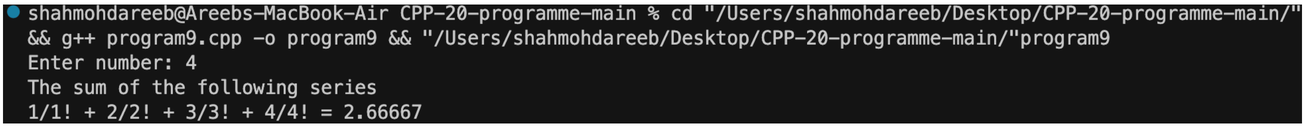
Theory:

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:

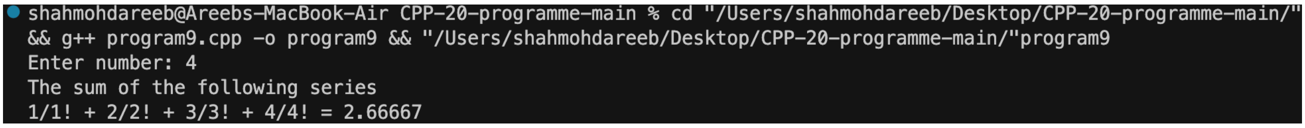
```
// sum of series 1/1! + 2/2! + 3/3! + ... + N/N! using nested
loops.
#include <iostream>
using namespace std;
// recursive function to find factorial of a number
int fact(int n) {
    return (n > 0) ? n * fact(n - 1) : 1;
}
// main function
int main() {
    // initlizing the sum and input a number
    int num;
    double sum = 0.0;
    cout << "Enter number: ";
    cin >> num;
    // calculating and displaying the sum  of the series
    cout << "The sum of the following series" << endl;
    for (int i = 1; i <= num; i++) {
        int nFact = 1;
        for (int j = 1; j <= i; j++) {
            nFact = nFact * j;
        }
        if (i < num) {
            cout << i << "/" << i << "! + ";
        } else {
            cout << i << "/" << i << "! = ";
        }
        sum += double(i) / nFact;
    }
    cout << sum << endl;
    return 0;
}
```

Output: (screenshot)

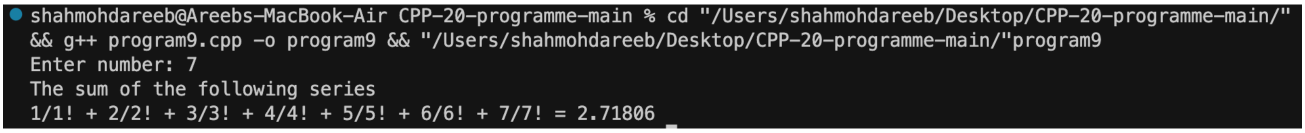


```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
&& g++ program9.cpp -o program9 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program9
Enter number: 4
The sum of the following series
1/1! + 2/2! + 3/3! + 4/4! = 2.66667
```

Test Case: Any two (screenshot)



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
&& g++ program9.cpp -o program9 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program9
Enter number: 4
The sum of the following series
1/1! + 2/2! + 3/3! + 4/4! = 2.66667
```



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
&& g++ program9.cpp -o program9 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program9
Enter number: 7
The sum of the following series
1/1! + 2/2! + 3/3! + 4/4! + 5/5! + 6/6! + 7/7! = 2.71806
```

Conclusion:

Hence, 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: **Shah Mohd Areeb Mohd Aslam**

Roll Number: **21**

Experiment No: 10

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

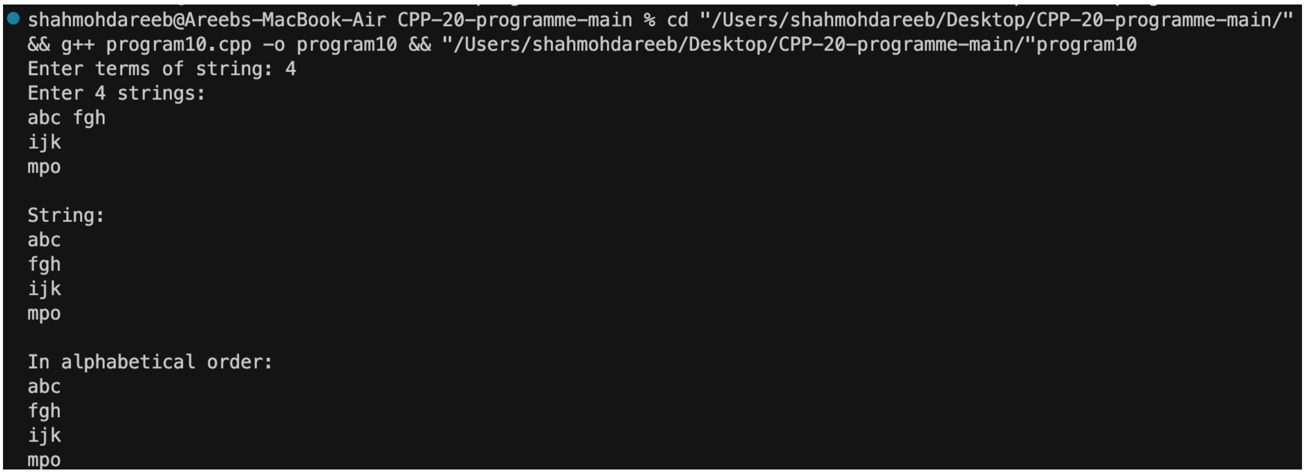
Theory:

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:

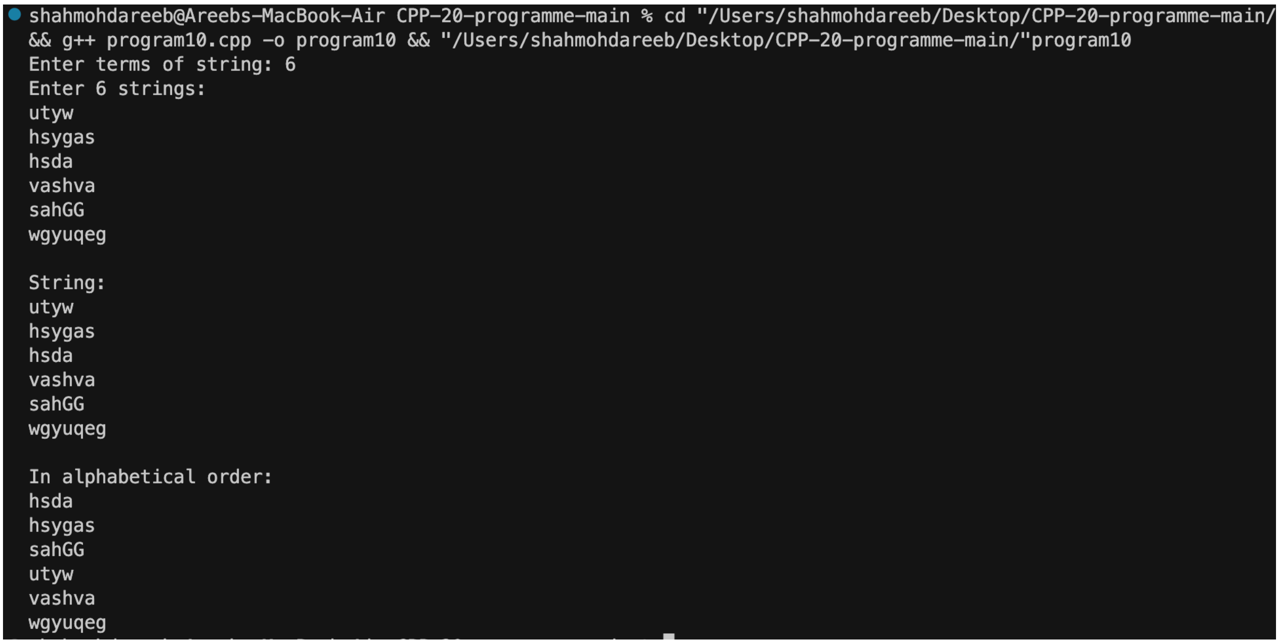
```
//program to create an array of strings and display them in alphabetical order
#include<iostream>
#include<string>
using namespace std;
int main()
{
    int n;
    cout<<"Enter terms of string: ";
    cin>>n;
    string str[n];
    cout<<"Enter "<<n<<" strings: "<<endl;
    for (int i=0; i<n; i++)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

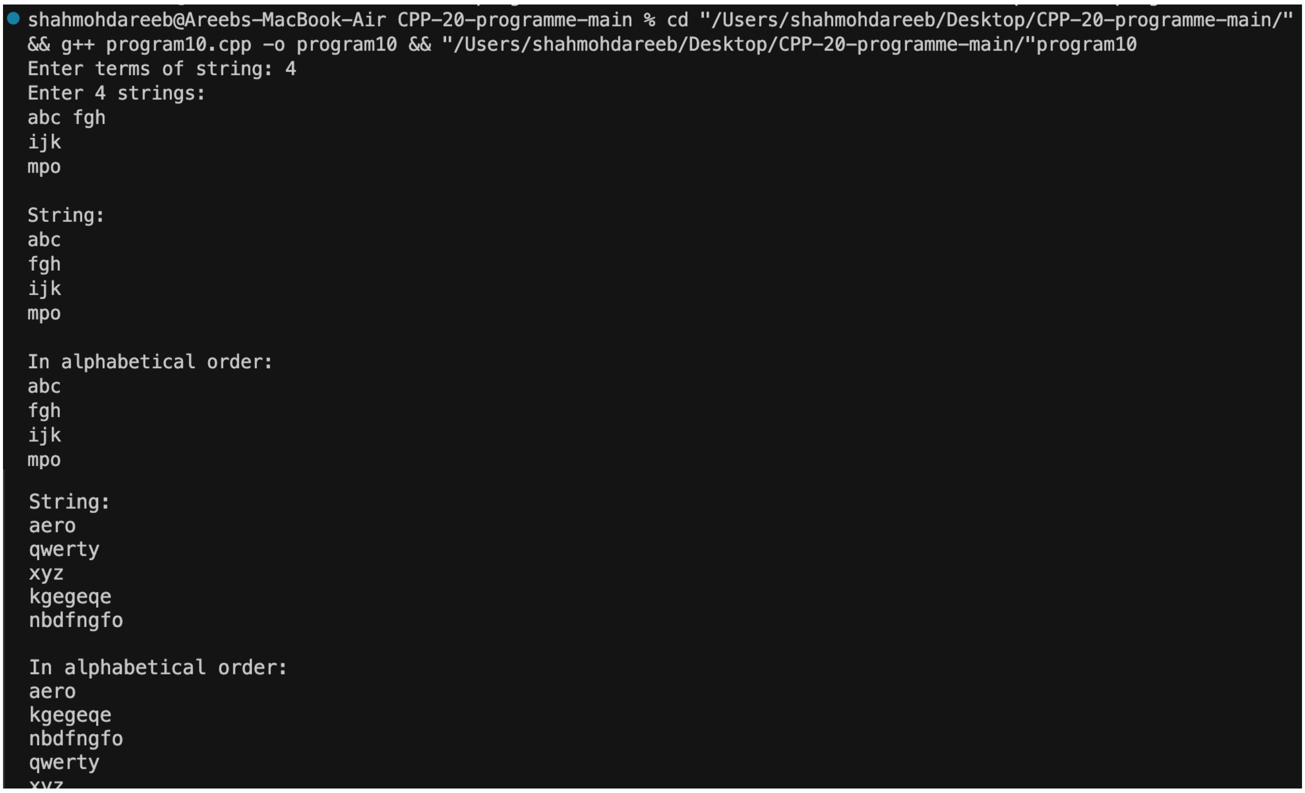


```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
&& g++ program10.cpp -o program10 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program10
Enter terms of string: 4
Enter 4 strings:
abc fgh
ijk
mpo

String:
abc
fgh
ijk
mpo

In alphabetical order:
abc
fgh
ijk
mpo
```

Test Case: Any two (screenshot)



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/
&& g++ program10.cpp -o program10 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program10
Enter terms of string: 6
Enter 6 strings:
utyw
hsygas
hsda
vashva
sahGG
wgyuqeg

String:
utyw
hsygas
hsda
vashva
sahGG
wgyuqeg

In alphabetical order:
hsda
hsygas
sahGG
utyw
vashva
wgyuqeg
```



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
&& g++ program10.cpp -o program10 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program10
Enter terms of string: 4
Enter 4 strings:
abc fgh
ijk
mpo

String:
abc
fgh
ijk
mpo

In alphabetical order:
abc
fgh
ijk
mpo

String:
aero
qwerty
xyz
kgegeqe
nbdfngfo

In alphabetical order:
aero
kgegeqe
nbdfngfo
qwerty
xyz
```

Conclusion:

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

Name of Student: **Shah Mohd Areeb Mohd Aslam**

Roll Number: **21**

Experiment No: 11

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

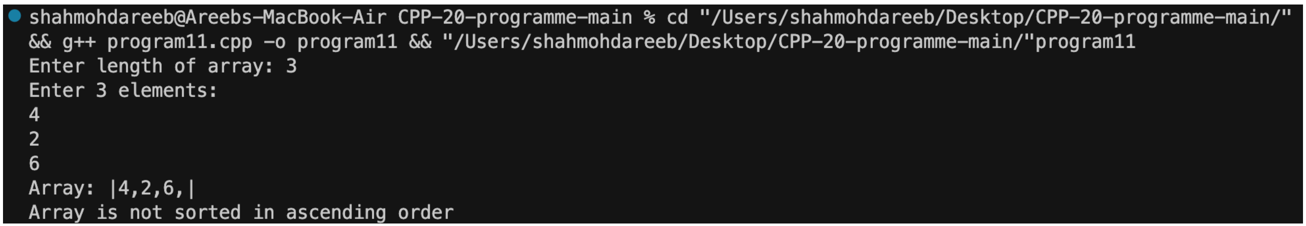
Theory:

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:

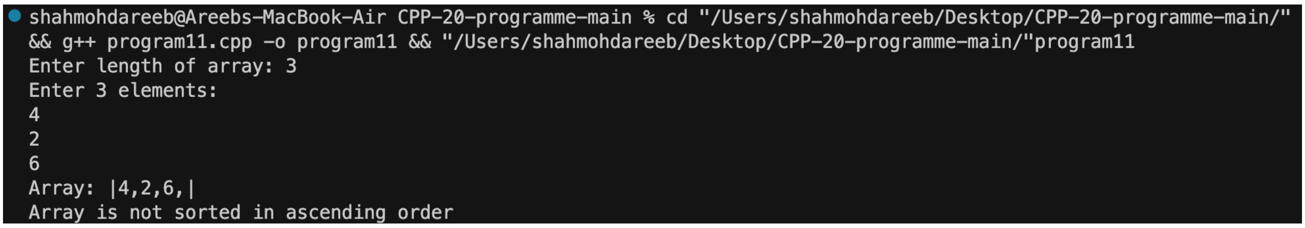
```
//program to check if an array is sorted in ascending order
#include<iostream>
using namespace std;
int main()
{
    int n;bool sorted = true;
    cout<<"Enter length of array: ";
    cin>>n;
    int arr[n];
    cout<<"Enter "<<n<<" elements: "<<endl;
    for (int i=0; i<n; i++)
    {
        cin>>arr[i];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

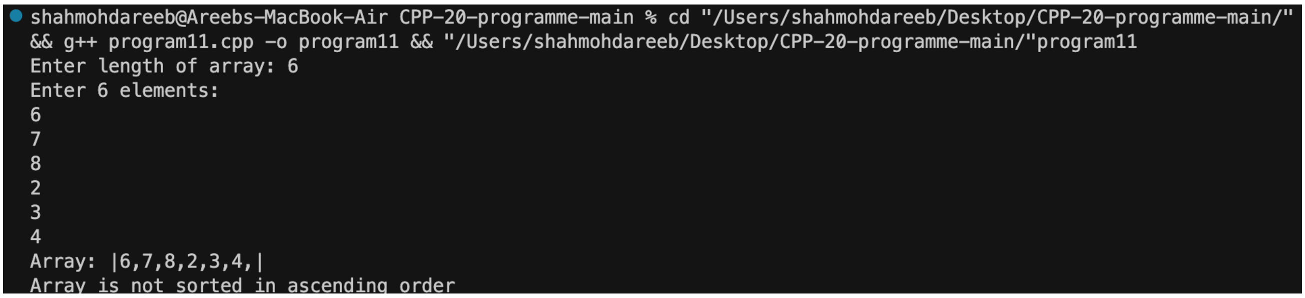


```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
&& g++ program11.cpp -o program11 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program11
Enter length of array: 3
Enter 3 elements:
4
2
6
Array: |4,2,6,|
Array is not sorted in ascending order
```

Test Case: Any two (screenshot)



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
 && g++ program11.cpp -o program11 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program11
 Enter length of array: 3
 Enter 3 elements:
 4
 2
 6
 Array: |4,2,6,|
 Array is not sorted in ascending order
```



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
 && g++ program11.cpp -o program11 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program11
 Enter length of array: 6
 Enter 6 elements:
 6
 7
 8
 2
 3
 4
 Array: |6,7,8,2,3,4,|
 Array is not sorted in ascending order
```

Conclusion:

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

Name of Student: Shah Mohd Areeb Mohd Aslam

Roll Number: 21

Experiment No: 12

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

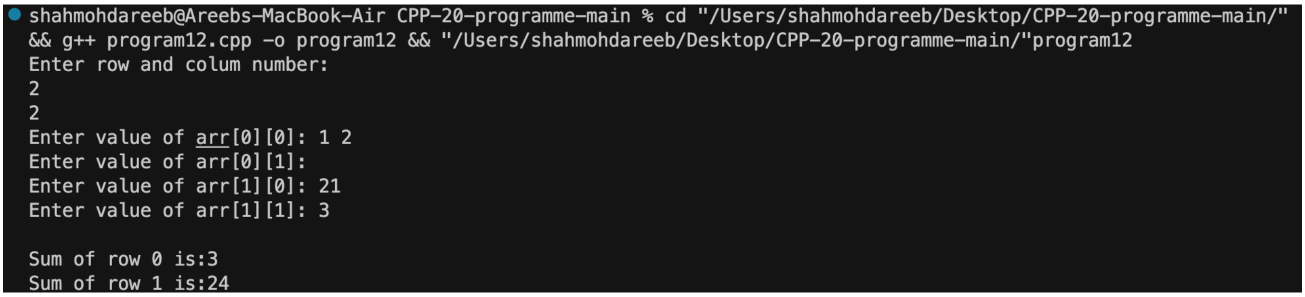
Theory:

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:

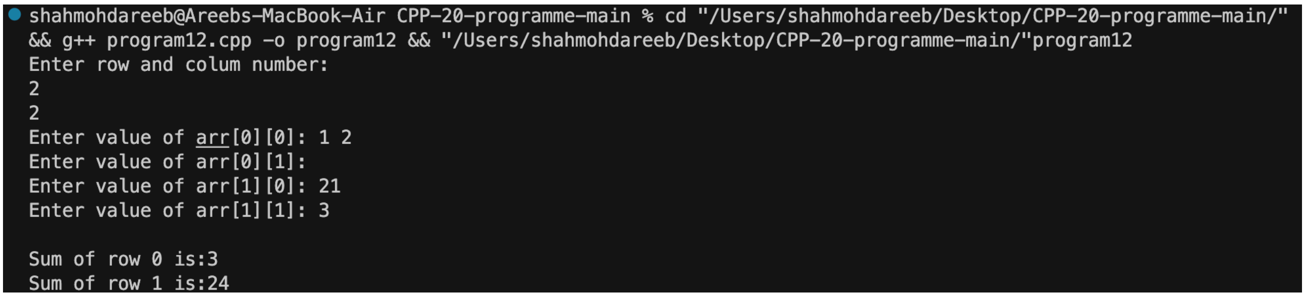
```
//Calculate the sum of elements in each row of a matrix.
#include <iostream>
using namespace std;
int main() {
    int rows, col;
    cout << "Enter row and colum number: " << endl;
    cin >> rows >> col;
    //creating a 2d array and entering elements in it
    int arr[rows][col];
    for(int i = 0; i < rows; i++){
        for(int j = 0; j < col; j++){
            cout << "Enter value of arr["<< i <<"]["<< j << "]: ";
            cin >> arr[i][j];
        }
        cout << endl;
    }
    //adding elements of each row in the sum variable
    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)

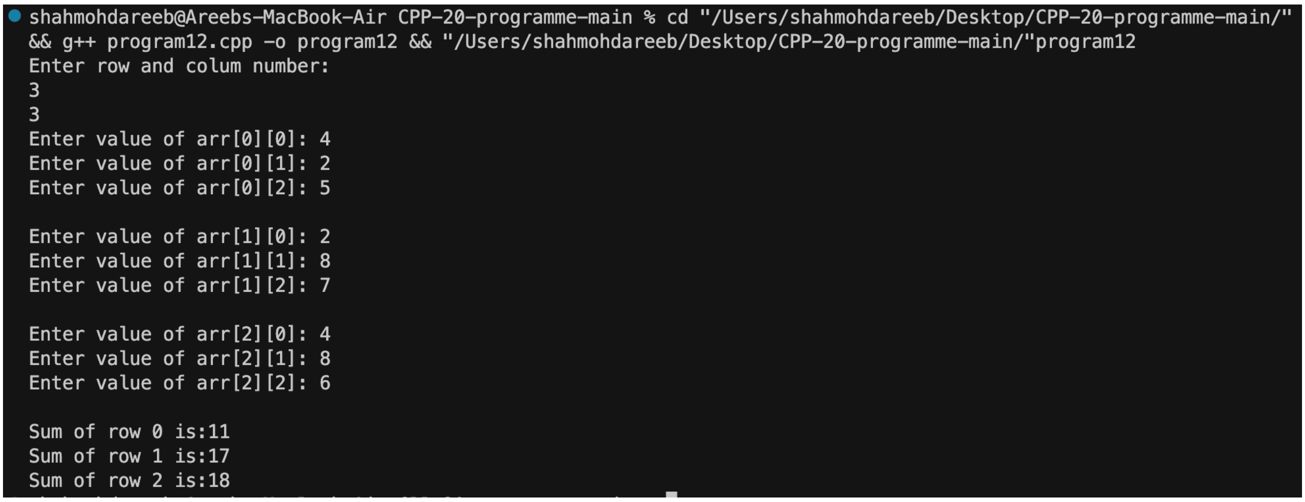


```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
&& g++ program12.cpp -o program12 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program12
Enter row and colum number:
2
2
Enter value of arr[0][0]: 1 2
Enter value of arr[0][1]:
Enter value of arr[1][0]: 21
Enter value of arr[1][1]: 3

Sum of row 0 is:3
Sum of row 1 is:24
```

Test Case: Any two (screenshot)



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
&& g++ program12.cpp -o program12 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program12
Enter row and colum number:
2
2
Enter value of arr[0][0]: 1 2
Enter value of arr[0][1]:
Enter value of arr[1][0]: 21
Enter value of arr[1][1]: 3

Sum of row 0 is:3
Sum of row 1 is:24
```



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
&& g++ program12.cpp -o program12 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program12
Enter row and colum number:
3
3
Enter value of arr[0][0]: 4
Enter value of arr[0][1]: 2
Enter value of arr[0][2]: 5

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

Enter value of arr[2][0]: 4
Enter value of arr[2][1]: 8
Enter value of arr[2][2]: 6

Sum of row 0 is:11
Sum of row 1 is:17
Sum of row 2 is:18
```

Conclusion:

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

Name of Student: Shah Mohd Areeb Mohd Aslam

Roll Number: 21

Experiment No: 13

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

Theory:

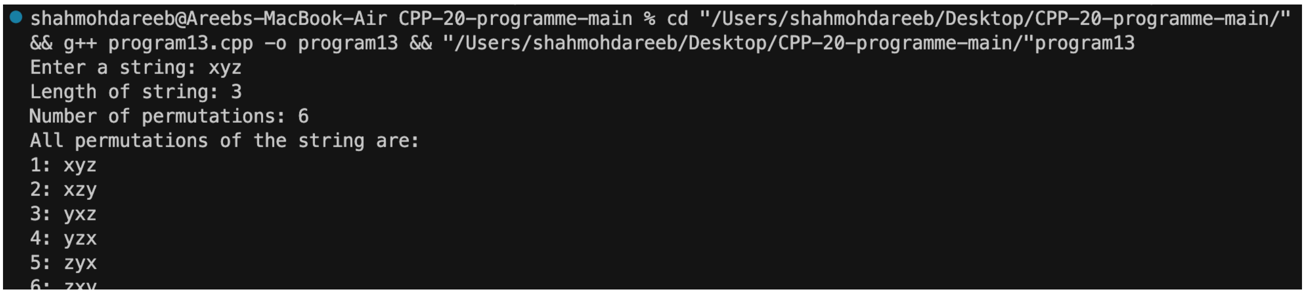
Make a factorial function to calculate factorial of a number(since there are as many permutations of a string as the factorial of number of characters in the string). Then making a permutation function to generate permutations of the string by fixing one character and swapping other characters and moving 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;
// function to calculate the factorial of a number
int fact(int a)
{
    if (a == 0 || a == 1)
    {
        return 1;
    }
    else
    {
        return a * fact(a - 1);
    }
}
// function to swap two characters in a string
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;
}
// function to generate permutations of a string
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++; // to print the permutation
            return;
        }
        return;
    }
    else
    {
        for (int i = left; i < right; ++i)
        {
            // fix the first character and recursively generate
permutations for the rest
            swap(str[left], str[i]);
            generatePermutations(str, left + 1, right, count,
words);
            // restore the string to its original state
(backtracking)
            swap(str[left], str[i]);
        }
    }
```

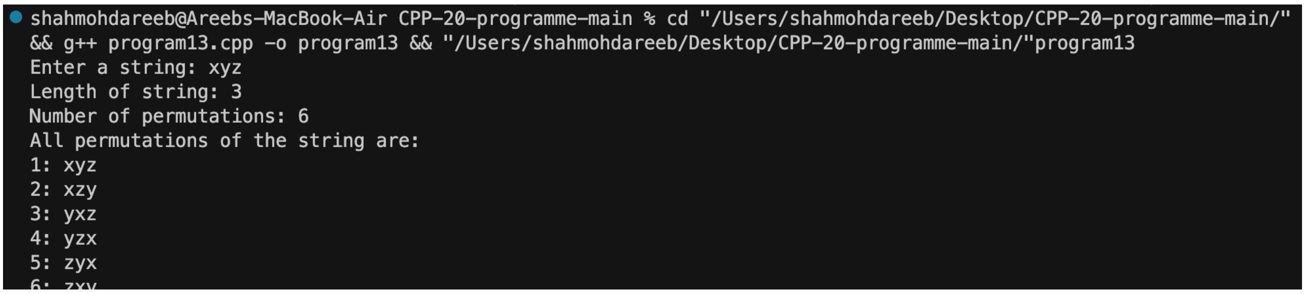
```
}
int main()
{
    string input;
    cout << "Enter a string: ";
    cin >> input;
    int n = input.length();
    cout << "Length of string: " << n << endl;
    int length = fact(n);
    cout << "Number of permutations: " << length << endl;
    cout << "All permutations of the string are:" << endl;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

Output: (screenshot)

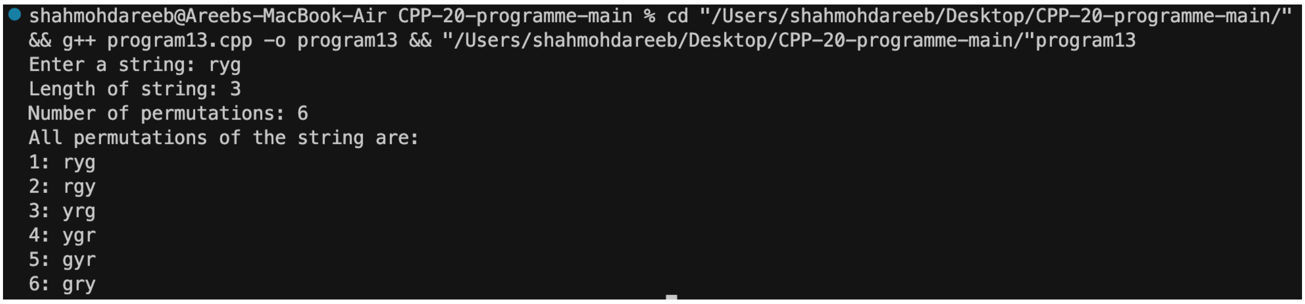


```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
&& g++ program13.cpp -o program13 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program13
Enter a string: xyz
Length of string: 3
Number of permutations: 6
All permutations of the string are:
1: xyz
2: xzy
3: yxz
4: yzx
5: zyx
6: zxy
```

Test Case: Any two (screenshot)



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
&& g++ program13.cpp -o program13 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program13
Enter a string: xyz
Length of string: 3
Number of permutations: 6
All permutations of the string are:
1: xyz
2: xzy
3: yxz
4: yzx
5: zyx
6: zxy
```



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
&& g++ program13.cpp -o program13 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program13
Enter a string: ryg
Length of string: 3
Number of permutations: 6
All permutations of the string are:
1: ryg
2: rgy
3: yrg
4: ygr
5: gyr
6: gry
```

Conclusion:

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

Name of Student: Shah Mohd Areeb Mohd Aslam

Roll Number: 21

Experiment No: 14

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

```
*****

* *

* *

* *

*****
```

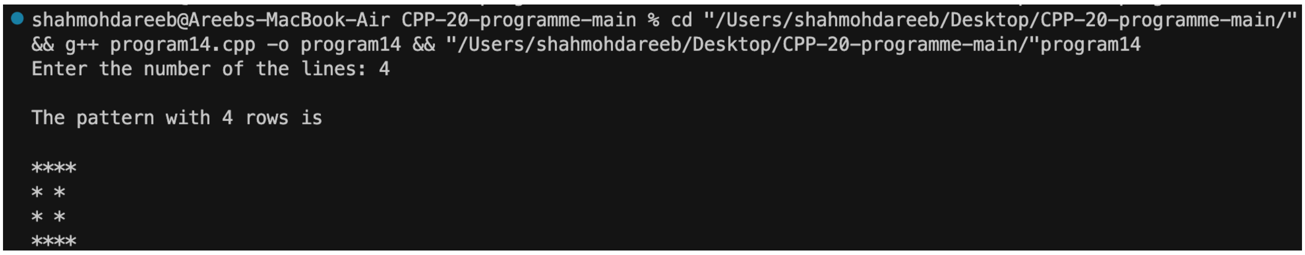
Theory:

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

Code:

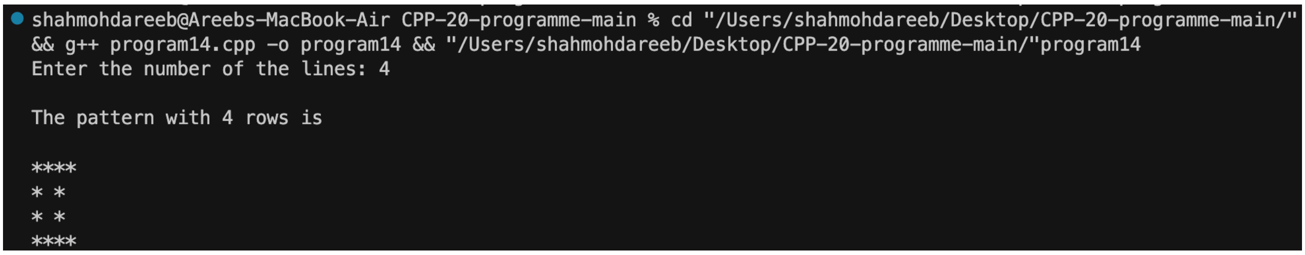
```cpp
// rectangular star pattern
#include <iostream>
using namespace std;
// main function
int main() {
    // input number of rows from user
    int num;
    cout << "Enter the number of the lines: ";
    cin >> num;
    // printting the pattern
    cout << endl << "The pattern with " << num << " 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)

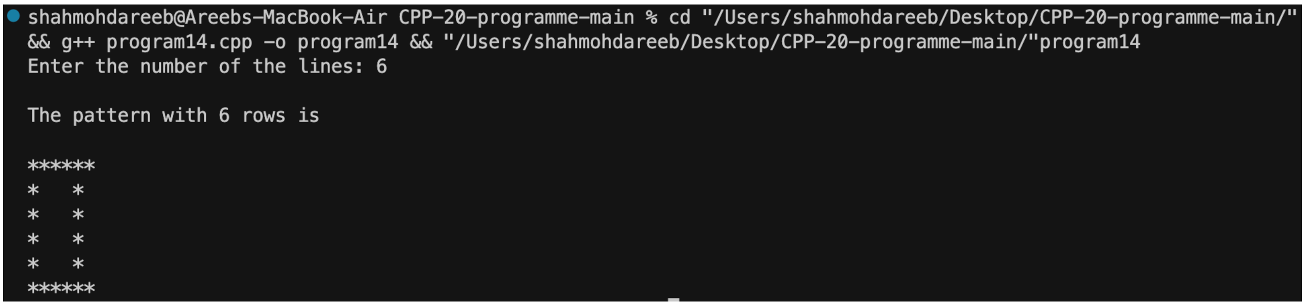


```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
&& g++ program14.cpp -o program14 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program14
Enter the number of the lines: 4

The pattern with 4 rows is

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

Test Case: Any two (screenshot)



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
&& g++ program14.cpp -o program14 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program14
Enter the number of the lines: 4

The pattern with 4 rows is

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



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
&& g++ program14.cpp -o program14 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program14
Enter the number of the lines: 6

The pattern with 6 rows is

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

Conclusion:

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

Name of Student: Shah Mohd Areeb Mohd Aslam

Roll Number: 21

Experiment No: 15

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

1

232

34543

4567654

34543

232

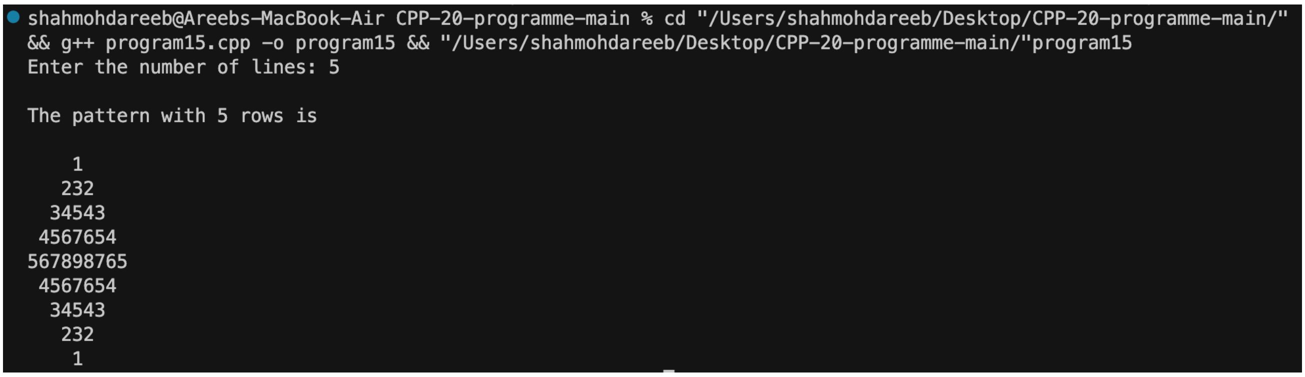
Theory:

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:

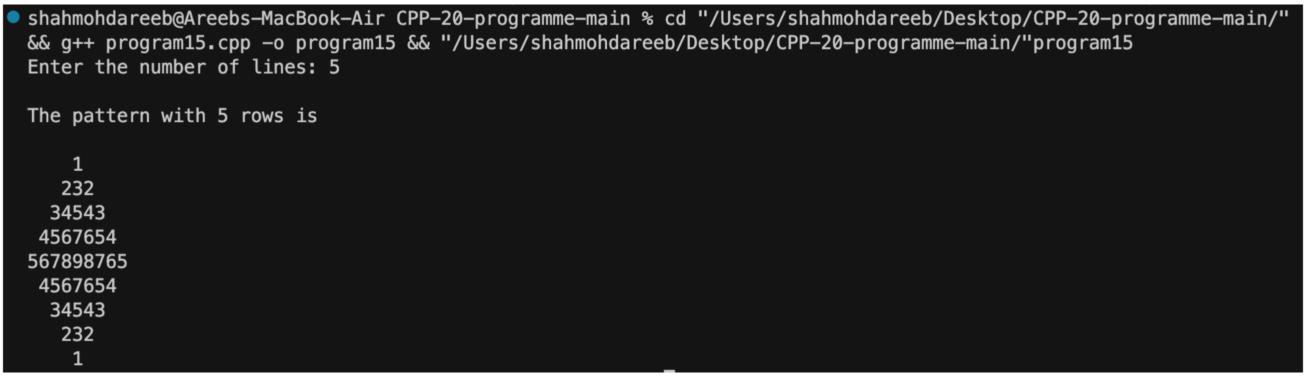
```
// pyramid number pattern
#include <iostream>
using namespace std;

// main function
int main() {
    // input number of rows from uesr
    int num;
    cout << "Enter the number of lines: ";
    cin >> num;

    // pritting pattern
    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)

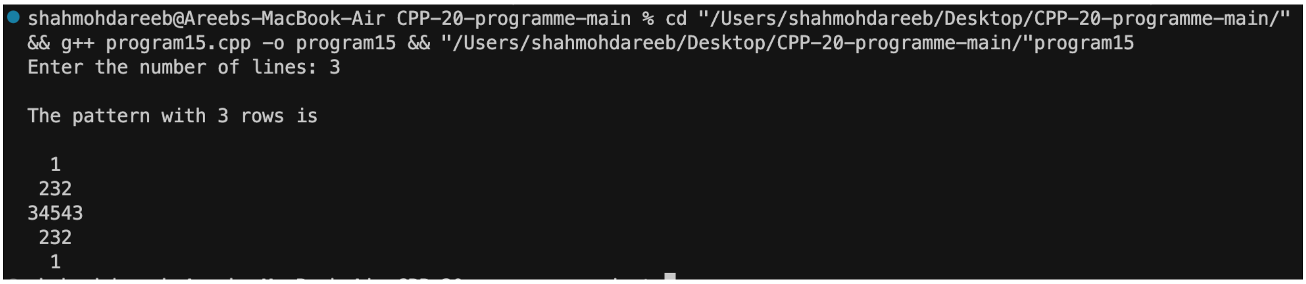


```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
&& g++ program15.cpp -o program15 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program15
Enter the number of lines: 5

The pattern with 5 rows is

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

Test Case: Any two (screenshot)



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
&& g++ program15.cpp -o program15 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program15
Enter the number of lines: 5

The pattern with 5 rows is

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



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
&& g++ program15.cpp -o program15 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program15
Enter the number of lines: 3

The pattern with 3 rows is

  1
 232
34543
 232
  1
```

Conclusion:

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

Name of Student: Shah Mohd Areeb Mohd Aslam

Roll Number: 21

Experiment No: 16

Title: Write a program to creating an inventory management system for a small store. The system should use object-oriented principles in C++. Your program should have the following features:

• Create a Product class that represents a product in the inventory. Each Product object should have the following attributes:

• Product ID (an integer)

• Product Name (a string)

• Price (a floating-point number)

• Quantity in stock (an integer)

• Implement a parameterized constructor for the Product class to initialize the attributes when a new product is added to the inventory.

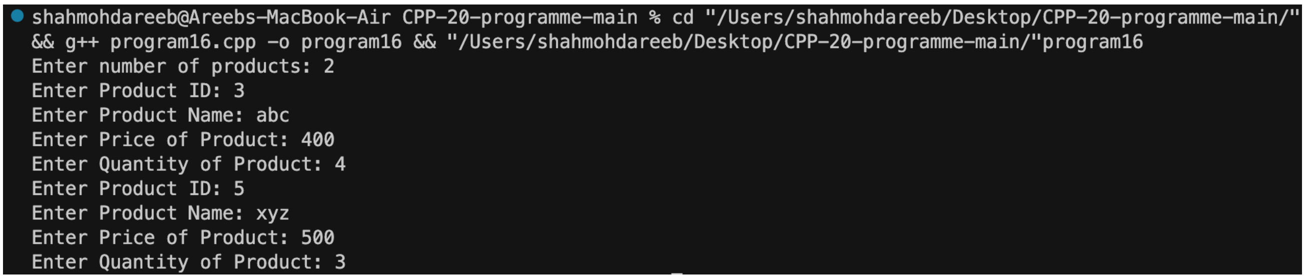
Theory:

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

Code:

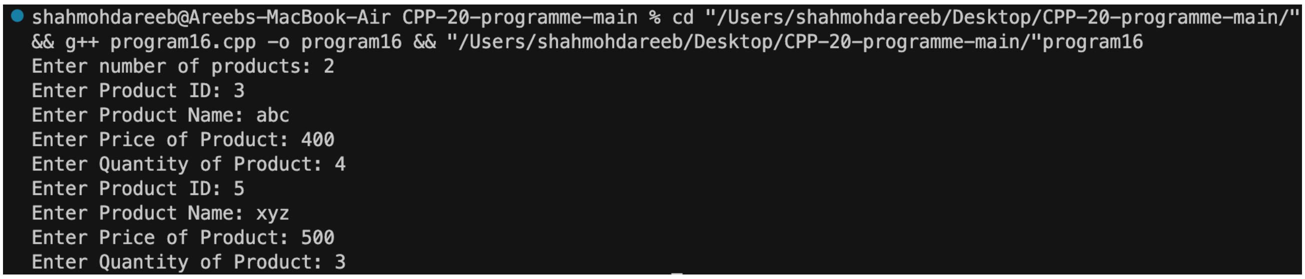
```
// store inventory management system
#include <iostream>
#include <string>
using namespace std;

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)

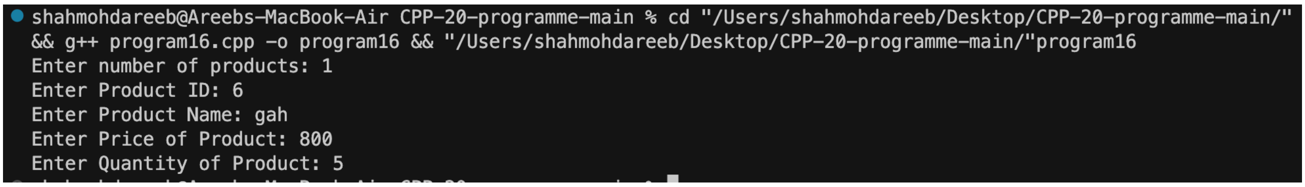


```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
&& g++ program16.cpp -o program16 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program16
Enter number of products: 2
Enter Product ID: 3
Enter Product Name: abc
Enter Price of Product: 400
Enter Quantity of Product: 4
Enter Product ID: 5
Enter Product Name: xyz
Enter Price of Product: 500
Enter Quantity of Product: 3
```

Test Case: Any two (screenshot)



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
&& g++ program16.cpp -o program16 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program16
Enter number of products: 2
Enter Product ID: 3
Enter Product Name: abc
Enter Price of Product: 400
Enter Quantity of Product: 4
Enter Product ID: 5
Enter Product Name: xyz
Enter Price of Product: 500
Enter Quantity of Product: 3
```



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
&& g++ program16.cpp -o program16 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program16
Enter number of products: 1
Enter Product ID: 6
Enter Product Name: gah
Enter Price of Product: 800
Enter Quantity of Product: 5
```

Conclusion:

Hence, 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: Shah Mohd Areeb Mohd Aslam

Roll Number: 21

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:

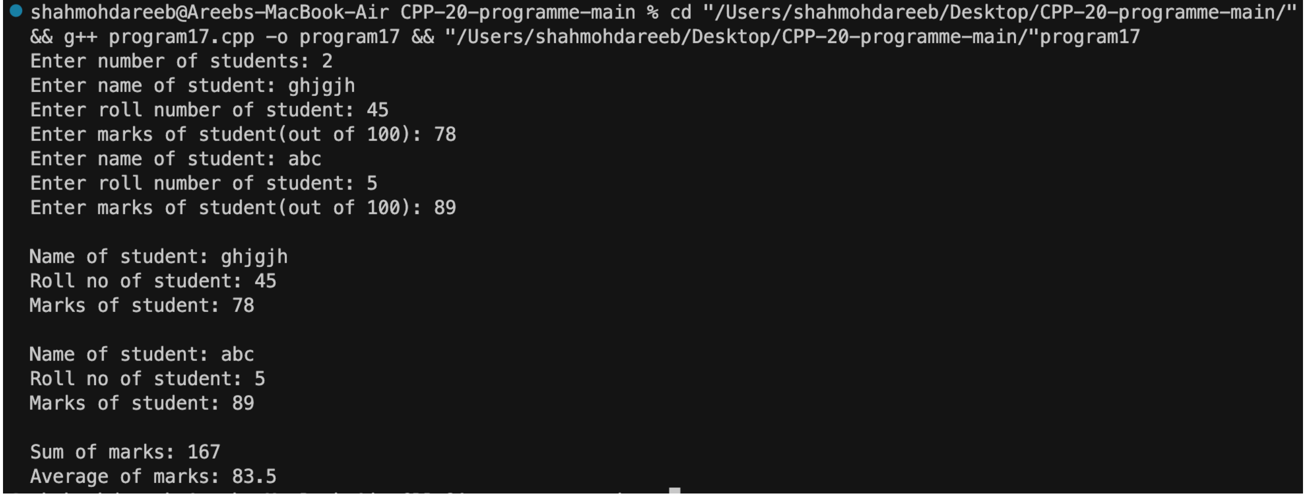
Using parameterised constructor to show details of students. Using for loop to take student details from the user and calculating average of students.

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

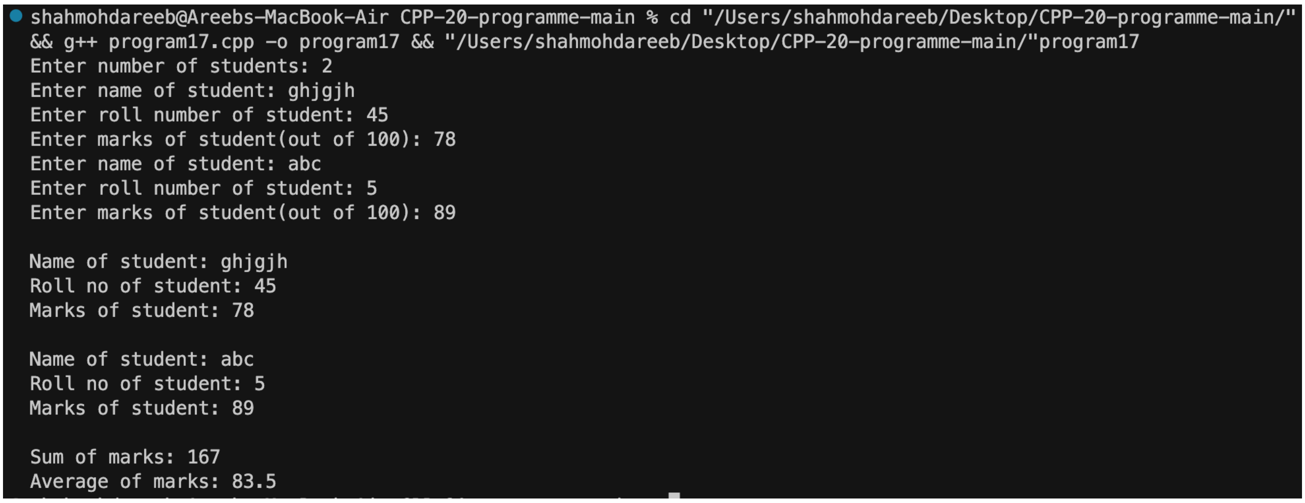
```
int main()
{
    int n, roll;
    float marks, sum=0;
    string name;
    cout << "Enter number of students: ";
    cin >> n;
    Student s[n];
    for (int i = 0; i < n; i++)
    {
        cout << "Enter name of student: ";
        cin.ignore();
        getline(cin, name);
        cout << "Enter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

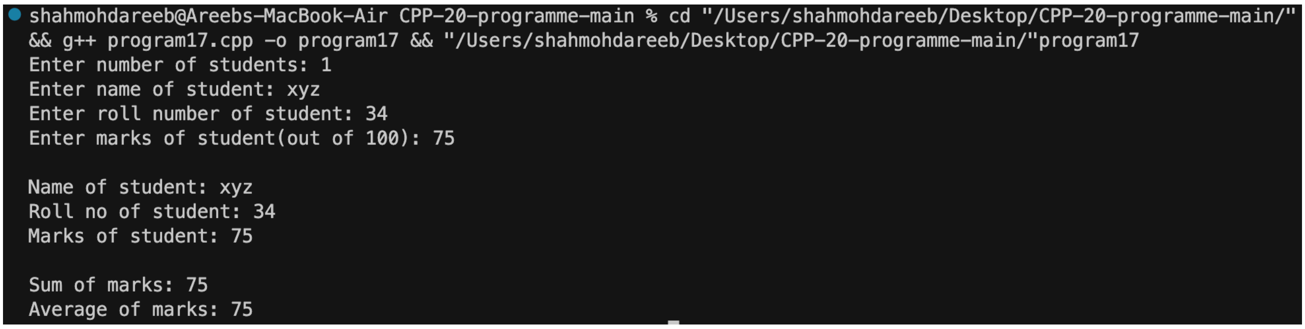


```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
 && g++ program17.cpp -o program17 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program17
 Enter number of students: 2
 Enter name of student: ghjgjh
 Enter roll number of student: 45
 Enter marks of student(out of 100): 78
 Enter name of student: abc
 Enter roll number of student: 5
 Enter marks of student(out of 100): 89

 Name of student: ghjgjh
 Roll no of student: 45
 Marks of student: 78

 Name of student: abc
 Roll no of student: 5
 Marks of student: 89

 Sum of marks: 167
 Average of marks: 83.5
```

Test Case: Any two (screenshot)



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
 && g++ program17.cpp -o program17 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program17
 Enter number of students: 2
 Enter name of student: ghjgjh
 Enter roll number of student: 45
 Enter marks of student(out of 100): 78
 Enter name of student: abc
 Enter roll number of student: 5
 Enter marks of student(out of 100): 89

 Name of student: ghjgjh
 Roll no of student: 45
 Marks of student: 78

 Name of student: abc
 Roll no of student: 5
 Marks of student: 89

 Sum of marks: 167
 Average of marks: 83.5
```



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
 && g++ program17.cpp -o program17 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program17
 Enter number of students: 1
 Enter name of student: xyz
 Enter roll number of student: 34
 Enter marks of student(out of 100): 75

 Name of student: xyz
 Roll no of student: 34
 Marks of student: 75

 Sum of marks: 75
 Average of marks: 75
```

Conclusion:

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

Name of Student: Shah Mohd Areeb Mohd Aslam

Roll Number: **21**

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.

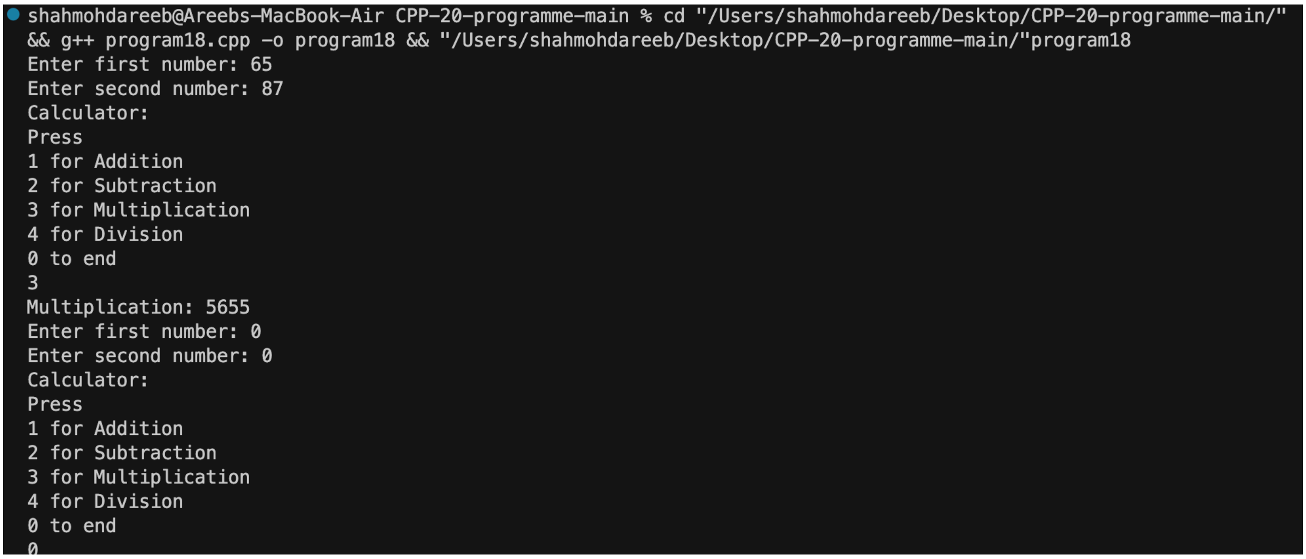
Theory:

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:

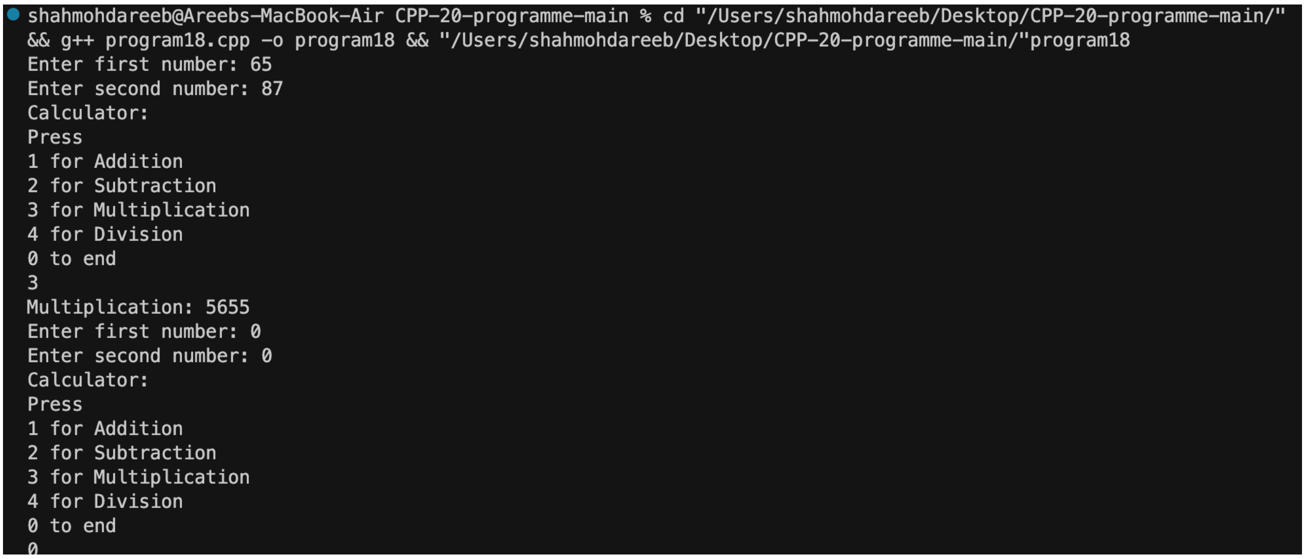
```
// program to simulate a simple calculator
#include <iostream>
using namespace std;

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)

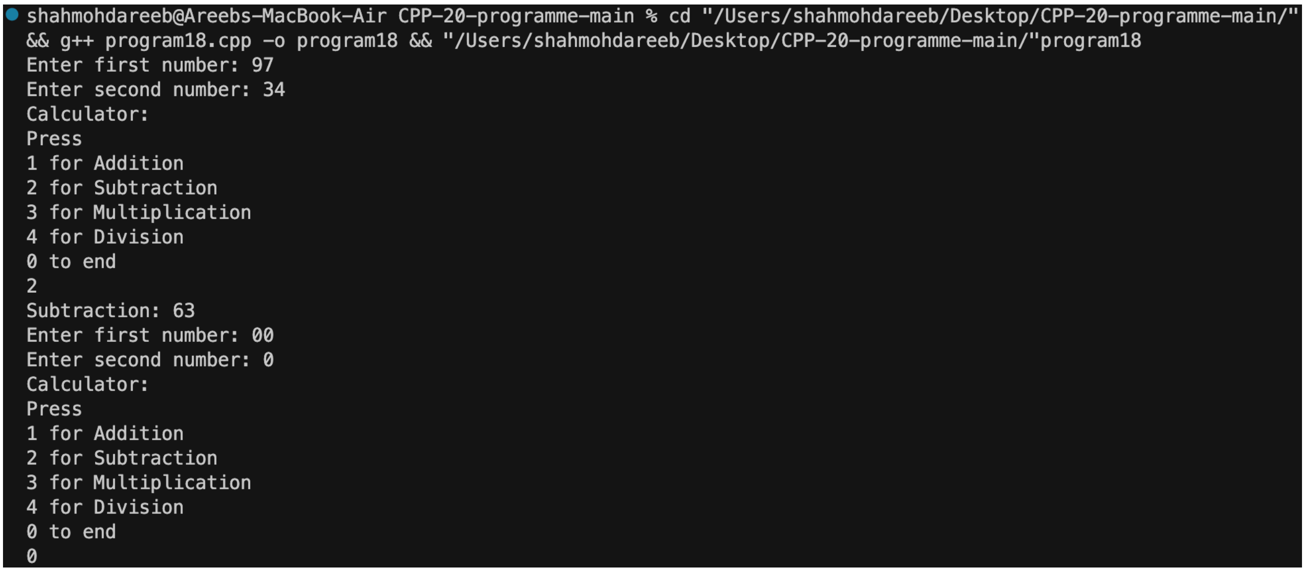


```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
&& g++ program18.cpp -o program18 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program18
Enter first number: 65
Enter second number: 87
Calculator:
Press
1 for Addition
2 for Subtraction
3 for Multiplication
4 for Division
0 to end
3
Multiplication: 5655
Enter first number: 0
Enter second number: 0
Calculator:
Press
1 for Addition
2 for Subtraction
3 for Multiplication
4 for Division
0 to end
0
```

Test Case: Any two (screenshot)



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
&& g++ program18.cpp -o program18 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program18
Enter first number: 65
Enter second number: 87
Calculator:
Press
1 for Addition
2 for Subtraction
3 for Multiplication
4 for Division
0 to end
3
Multiplication: 5655
Enter first number: 0
Enter second number: 0
Calculator:
Press
1 for Addition
2 for Subtraction
3 for Multiplication
4 for Division
0 to end
0
```



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
&& g++ program18.cpp -o program18 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program18
Enter first number: 97
Enter second number: 34
Calculator:
Press
1 for Addition
2 for Subtraction
3 for Multiplication
4 for Division
0 to end
2
Subtraction: 63
Enter first number: 00
Enter second number: 0
Calculator:
Press
1 for Addition
2 for Subtraction
3 for Multiplication
4 for Division
0 to end
0
```

Conclusion:

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

Name of Student: Shah Mohd Areeb Mohd Aslam

Roll Number: 21

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.

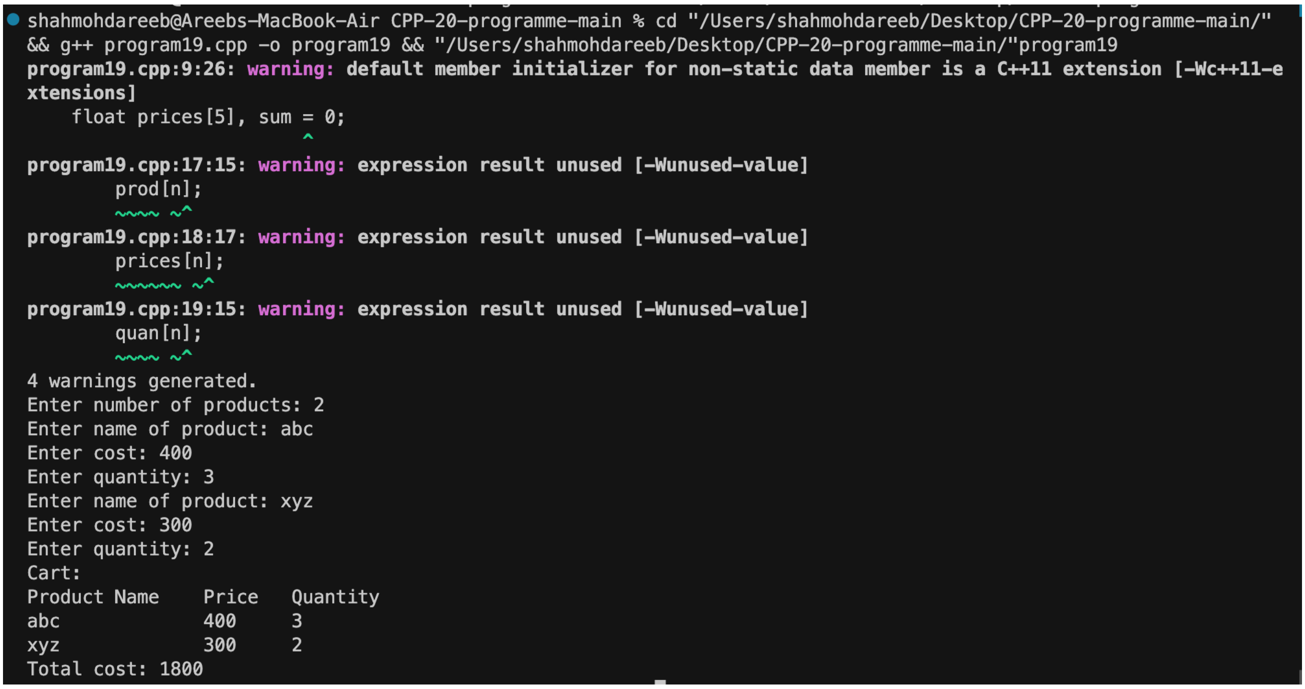
Theory:

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:

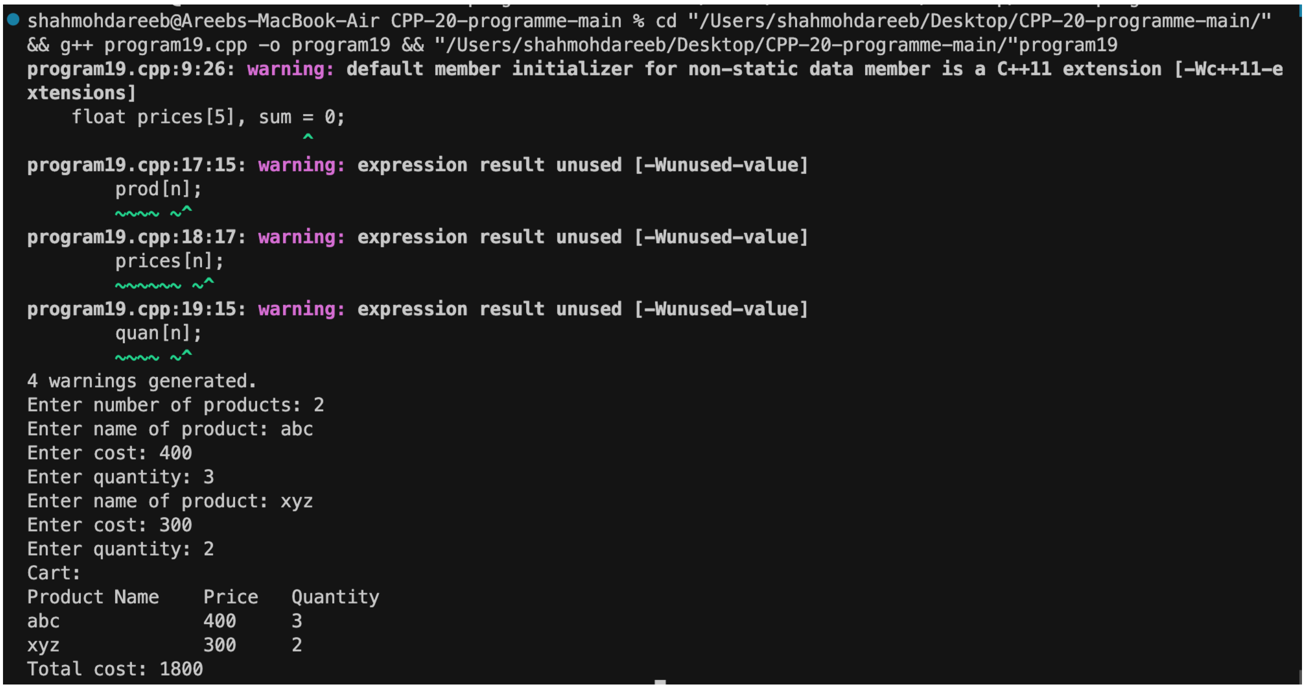
```
// online shop simulator
#include <iostream>
using namespace std;

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)

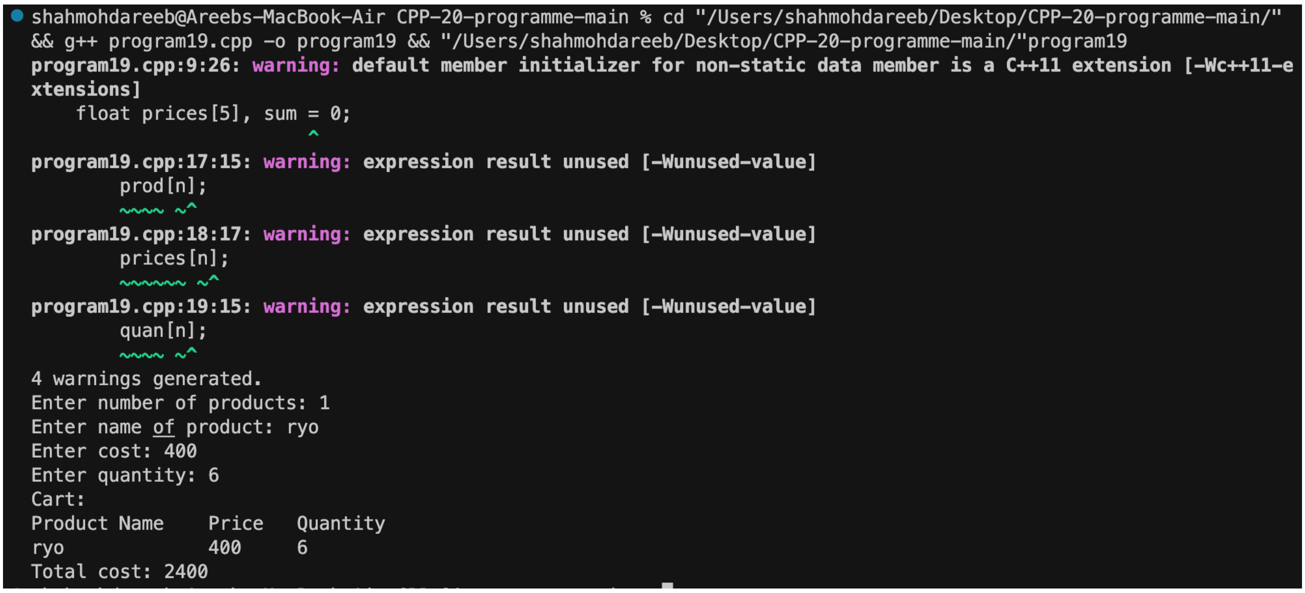


```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
&& g++ program19.cpp -o program19 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program19
program19.cpp:9:26: warning: default member initializer for non-static data member is a C++11 extension [-Wc++11-e
xtensions]
    float prices[5], sum = 0;
                         ^
program19.cpp:17:15: warning: expression result unused [-Wunused-value]
        prod[n];
        ~~~~ ~^
program19.cpp:18:17: warning: expression result unused [-Wunused-value]
        prices[n];
        ~~~~~~ ~^
program19.cpp:19:15: warning: expression result unused [-Wunused-value]
        quan[n];
        ~~~~ ~^
4 warnings generated.
Enter number of products: 2
Enter name of product: abc
Enter cost: 400
Enter quantity: 3
Enter name of product: xyz
Enter cost: 300
Enter quantity: 2
Cart:
Product Name    Price   Quantity
abc             400     3
xyz             300     2
Total cost: 1800
```

Test Case: Any two (screenshot)



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
&& g++ program19.cpp -o program19 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program19
program19.cpp:9:26: warning: default member initializer for non-static data member is a C++11 extension [-Wc++11-e
xtensions]
    float prices[5], sum = 0;
                         ^
program19.cpp:17:15: warning: expression result unused [-Wunused-value]
        prod[n];
        ~~~~ ~^
program19.cpp:18:17: warning: expression result unused [-Wunused-value]
        prices[n];
        ~~~~~~ ~^
program19.cpp:19:15: warning: expression result unused [-Wunused-value]
        quan[n];
        ~~~~ ~^
4 warnings generated.
Enter number of products: 2
Enter name of product: abc
Enter cost: 400
Enter quantity: 3
Enter name of product: xyz
Enter cost: 300
Enter quantity: 2
Cart:
Product Name    Price   Quantity
abc             400     3
xyz             300     2
Total cost: 1800
```



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
&& g++ program19.cpp -o program19 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program19
program19.cpp:9:26: warning: default member initializer for non-static data member is a C++11 extension [-Wc++11-e
xtensions]
    float prices[5], sum = 0;
                         ^
program19.cpp:17:15: warning: expression result unused [-Wunused-value]
        prod[n];
        ~~~~ ~^
program19.cpp:18:17: warning: expression result unused [-Wunused-value]
        prices[n];
        ~~~~~~ ~^
program19.cpp:19:15: warning: expression result unused [-Wunused-value]
        quan[n];
        ~~~~ ~^
4 warnings generated.
Enter number of products: 1
Enter name of product: ryo
Enter cost: 400
Enter quantity: 6
Cart:
Product Name    Price   Quantity
ryo             400     6
Total cost: 2400
```

Conclusion:

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

Name of Student: Shah Mohd Areeb Mohd Aslam

Roll Number: 21

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.

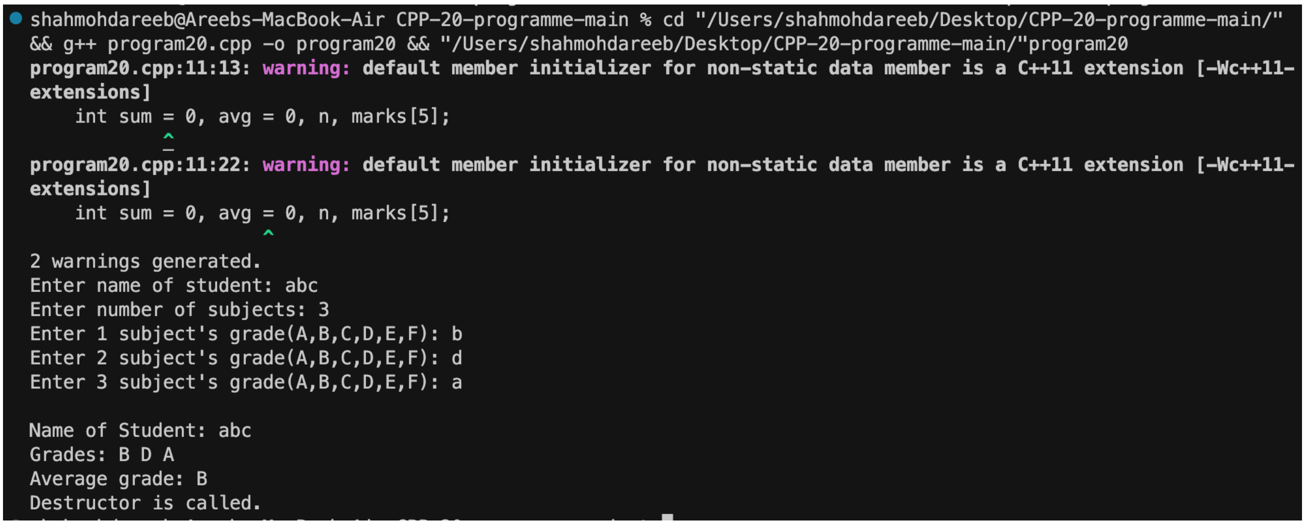
Code:

```
// program to manage grades of students in a classroom
#include <iostream>
#include <string>
using namespace std;

class Student
{
```

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

public:
    Student()
    {
        cout << "Enter name of student: ";
        getline(cin, name);
    }
    void addGrade()
    {
        cout << "Enter number of subjects: ";
        cin >> n;
        for (int i = 0; i < n; i++)
        {
            cout << "Enter " << i + 1 << " subject's grade(A,B,C,D,E,F): ";
            cin >> grade[i];
        }
    }
    void averageGrade()
    {
        for (int i = 0; i < n; i++)
        {
            if (tolower(grade[i]) == 'a')
            {
                marks[i] = 100;
            }
            else if (tolower(grade[i]) == 'b')
            {
                marks[i] = 90;
            }
            else if (tolower(grade[i]) == 'c')
            {
                marks[i] = 80;
            }
            else if (tolower(grade[i]) == 'd')
            {
                marks[i] = 70;
            }
            else if (tolower(grade[i]) == 'e')
            {
                marks[i] = 60;
            }
            else
            {
                marks[i] = 50;
            }
        }
        for (int i = 0; i < n; i++)
        {
            sum += marks[i];
        }
        avg = sum / n;
        if (avg > 90)
        {
            grades = 'A';
        }
        else if (avg > 80 && avg <= 90)
        {
            grades = 'B';
        }
        else if (avg > 70 && avg <= 80)
        {
            grades = 'C';
        }
        else if (avg > 60 && avg <= 70)
        {
            grades = 'D';
        }
        else if (avg > 50 && avg <= 60)
        {
            grades = 'E';
        }
        else
        {
            grades = 'F';
        }
    }
    void showDetails()
    {
        cout << endl
             << "Name of Student: " << name << endl;
        cout << "Grades: ";
        for (int i = 0; i < n; i++)
        {
            cout << (char)toupper(grade[i]) << " ";
        }
        cout << endl
             << "Average grade: " << grades << endl;
    }
    ~Student()
    {
        cout << "Destructor is called." << endl;
    }
};
```

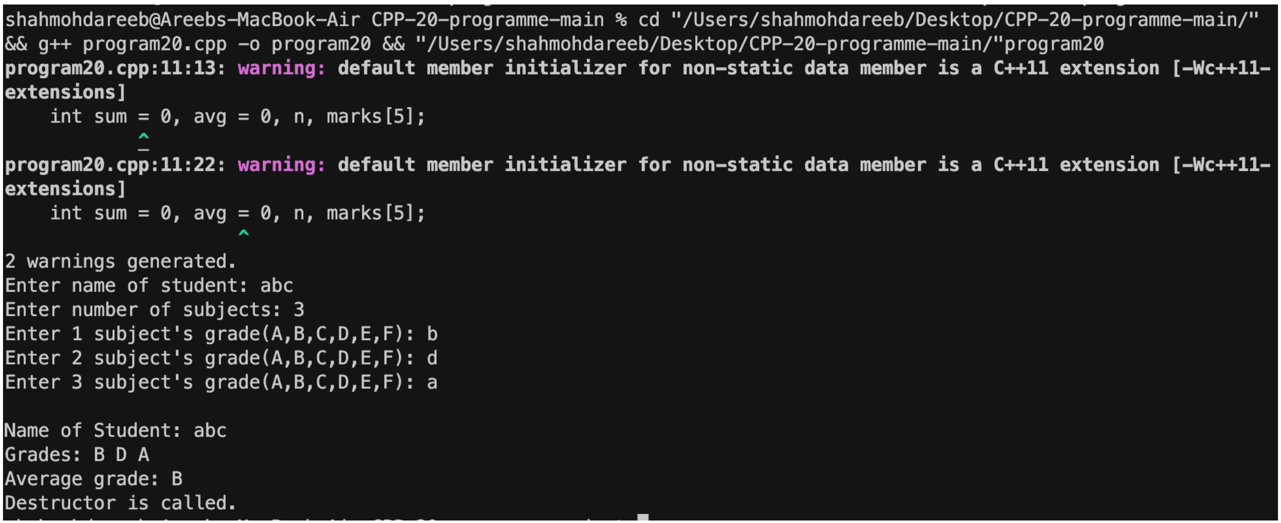
```
int main()
{
    Student s1;
    s1.addGrade();
    s1.averageGrade();
    s1.showDetails();
    return 0;
}
```

Output: (screenshot)

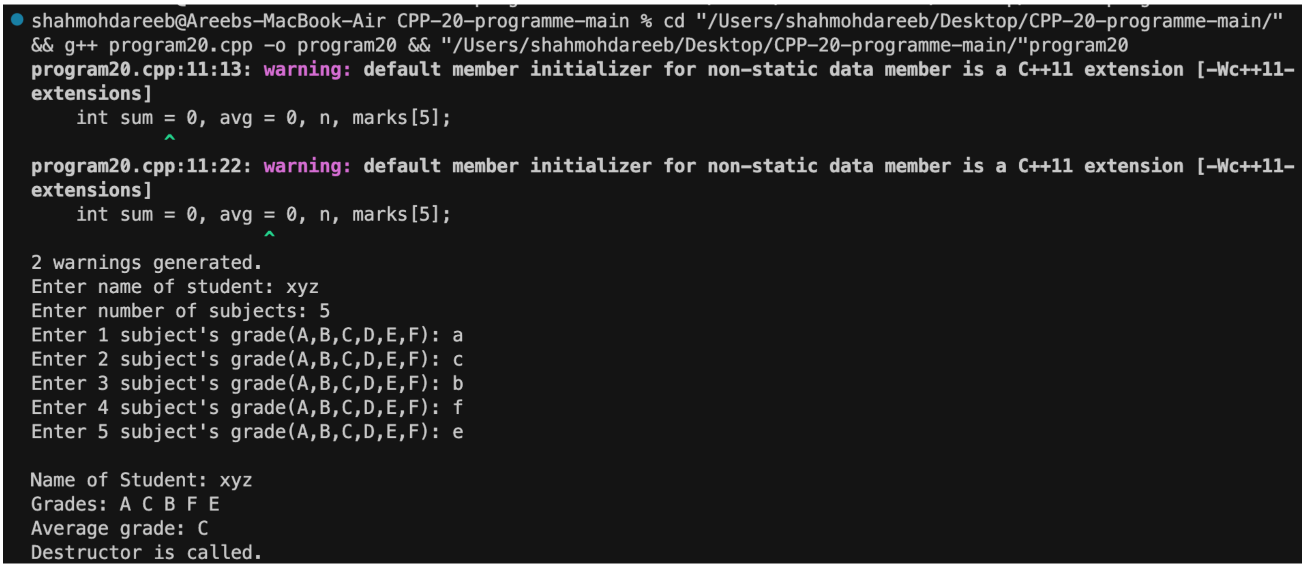


```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
&& g++ program20.cpp -o program20 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program20
program20.cpp:11:13: warning: default member initializer for non-static data member is a C++11 extension [-Wc++11-
extensions]
    int sum = 0, avg = 0, n, marks[5];
            ^
            _
program20.cpp:11:22: warning: default member initializer for non-static data member is a C++11 extension [-Wc++11-
extensions]
    int sum = 0, avg = 0, n, marks[5];
                     ^

2 warnings generated.
Enter name of student: abc
Enter number of subjects: 3
Enter 1 subject's grade(A,B,C,D,E,F): b
Enter 2 subject's grade(A,B,C,D,E,F): d
Enter 3 subject's grade(A,B,C,D,E,F): a

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

Test Case: Any two (screenshot):



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
&& g++ program20.cpp -o program20 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program20
program20.cpp:11:13: warning: default member initializer for non-static data member is a C++11 extension [-Wc++11-
extensions]
    int sum = 0, avg = 0, n, marks[5];
            ^
            _
program20.cpp:11:22: warning: default member initializer for non-static data member is a C++11 extension [-Wc++11-
extensions]
    int sum = 0, avg = 0, n, marks[5];
                     ^

2 warnings generated.
Enter name of student: abc
Enter number of subjects: 3
Enter 1 subject's grade(A,B,C,D,E,F): b
Enter 2 subject's grade(A,B,C,D,E,F): d
Enter 3 subject's grade(A,B,C,D,E,F): a

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



```
shahmohdareeb@Areebs-MacBook-Air CPP-20-programme-main % cd "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"
&& g++ program20.cpp -o program20 && "/Users/shahmohdareeb/Desktop/CPP-20-programme-main/"program20
program20.cpp:11:13: warning: default member initializer for non-static data member is a C++11 extension [-Wc++11-
extensions]
    int sum = 0, avg = 0, n, marks[5];
            ^
program20.cpp:11:22: warning: default member initializer for non-static data member is a C++11 extension [-Wc++11-
extensions]
    int sum = 0, avg = 0, n, marks[5];
                     ^
2 warnings generated.
Enter name of student: xyz
Enter number of subjects: 5
Enter 1 subject's grade(A,B,C,D,E,F): a
Enter 2 subject's grade(A,B,C,D,E,F): c
Enter 3 subject's grade(A,B,C,D,E,F): b
Enter 4 subject's grade(A,B,C,D,E,F): f
Enter 5 subject's grade(A,B,C,D,E,F): e

Name of Student: xyz
Grades: A C B F E
Average grade: C
Destructor is called.
```

Conclusion:

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