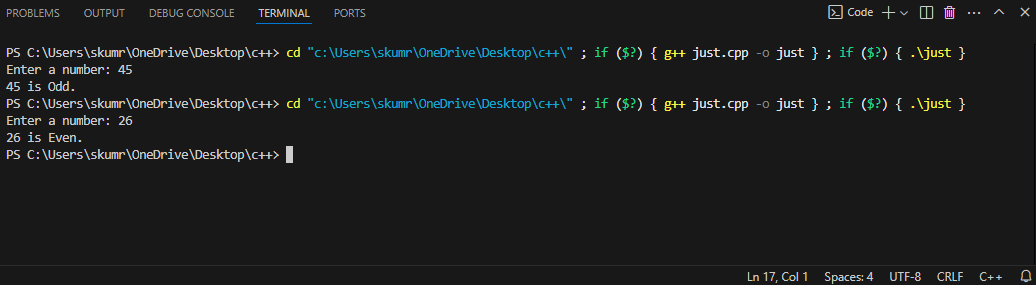
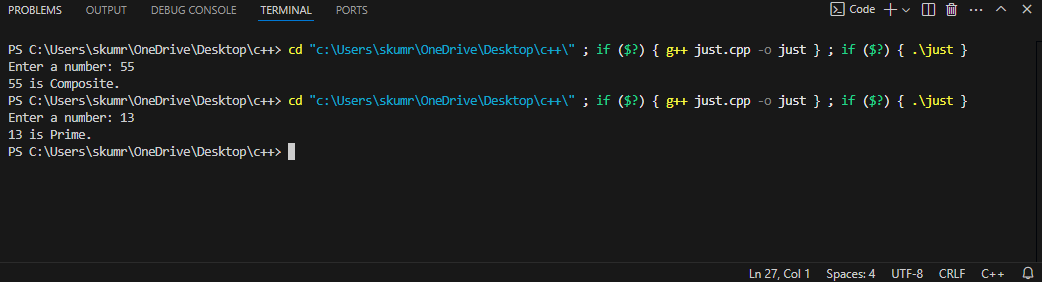
**1. Write a program to check whether a number is even or odd.**

#include <iostream>  
using namespace std;  
  
int main() {  
 int num;  
 cout << "Enter a number: ";  
 cin >> num;  
   
 if (num % 2 == 0) {  
 cout << num << " is Even." << endl;  
 } else {  
 cout << num << " is Odd." << endl;  
 }  
   
 return 0;  
}



**2. Write a program to check whether a number is prime or composite.**

#include <iostream>  
using namespace std;  
  
int main() {  
 int num, i, flag = 0;  
 cout << "Enter a number: ";  
 cin >> num;  
  
 if (num <= 1) {  
 cout << num << " is neither prime nor composite." << endl;  
 } else {  
 for (i = 2; i <= num / 2; i++) {  
 if (num % i == 0) {  
 flag = 1;  
 break;  
 }  
 }  
 if (flag == 0) {  
 cout << num << " is Prime." << endl;  
 } else {  
 cout << num << " is Composite." << endl;  
 }  
 }  
  
 return 0;  
}



**3. Write a program to print a table of a given number up to n multiples.**

#include <iostream>  
using namespace std;  
  
int main() {  
 int num, n;  
 cout << "Enter a number: ";  
 cin >> num;  
 cout << "Enter the limit (n): ";  
 cin >> n;  
  
 for (int i = 1; i <= n; i++) {  
 cout << num << " x " << i << " = " << num \* i << endl;  
 }  
  
 return 0;  
}



**4.Write a program**

**I].greater of two numbers**

#include <iostream>

using namespace std;

int main() {

    int num1, num2;

    cout << "Enter two numbers: ";

    cin >> num1 >> num2;

    if (num1 > num2) {

        cout << "The greater number is: " << num1 << endl;

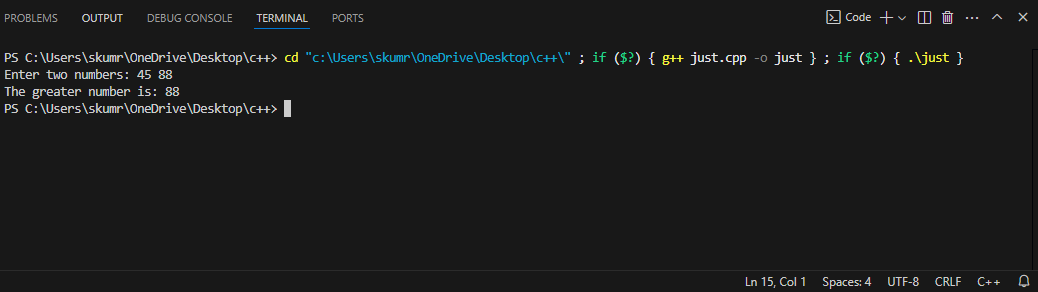
    } else {

        cout << "The greater number is: " << num2 << endl;

    }

    return 0;

}



**II]greater of three numbers.**

#include <iostream>

using namespace std;

int main() {

    int num1, num2, num3;

    cout << "Enter three numbers: ";

    cin >> num1 >> num2 >> num3;

    if (num1 >= num2 && num1 >= num3) {

        cout << "The greatest number is: " << num1 << endl;

    } else if (num2 >= num1 && num2 >= num3) {

        cout << "The greatest number is: " << num2 << endl;

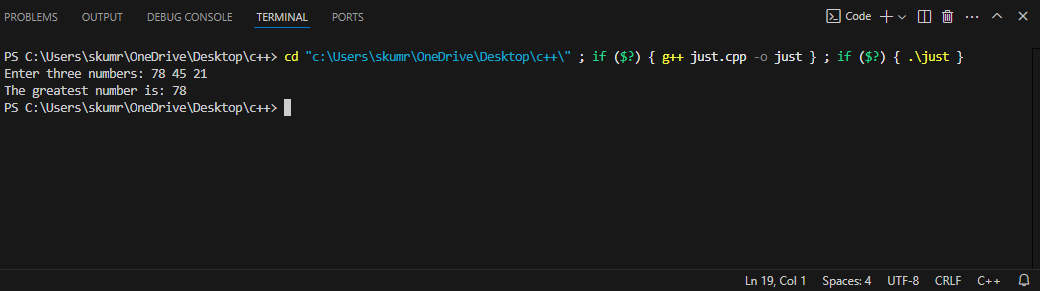
    } else {

        cout << "The greatest number is: " << num3 << endl;

    }

    return 0;

}



**5.Write a program to find sum of first “n” natural numbers.**

#include <iostream>

using namespace std;

int main() {

    int n, sum = 0;

    cout << "Enter a number: ";

    cin >> n;

    for (int i = 1; i <= n; i++) {

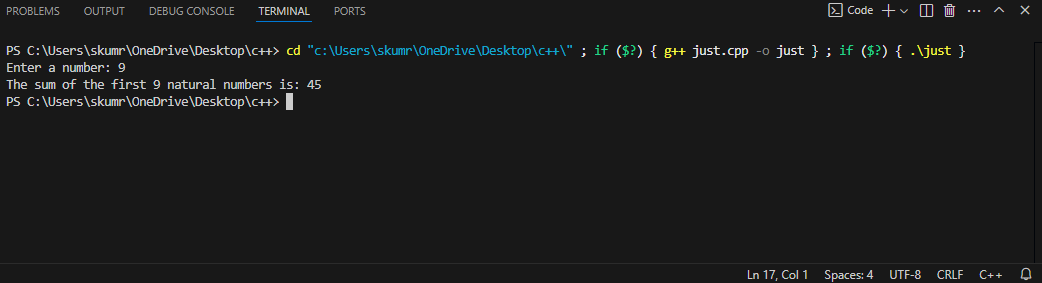
        sum += i;

    }

    cout << "The sum of the first " << n << " natural numbers is: " << sum << endl;

    return 0;

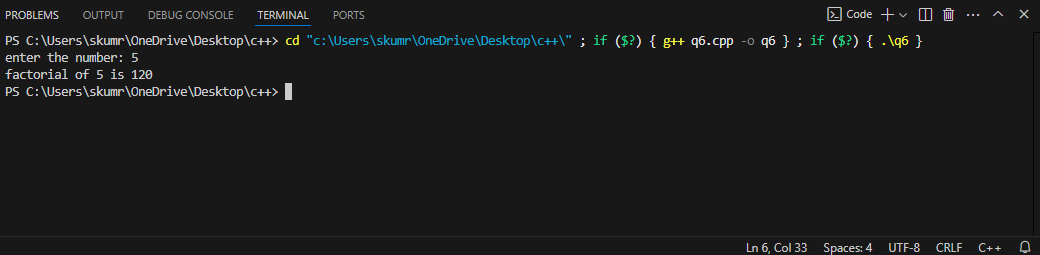
}

****

**6. Write a program to find the factorial of a number.**

#include <iostream>  
using namespace std;  
int main(){  
 int n;  
 int pro=1;  
 cout << "Enter the number: ";  
 cin >> n;  
 for (int i=1; i<=n; i++){  
 pro = pro \* i;  
 }  
 cout << "Factorial of " << n << " is " << pro;  
 return 0;  
}

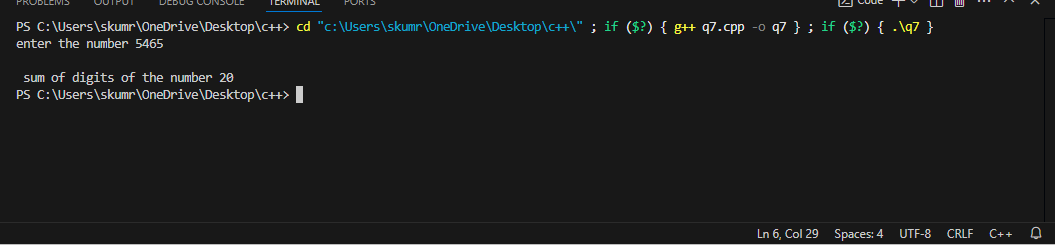
Output for Factorial Program:



**7. Write a program to calculate the sum of digits of a number.**

#include <iostream>  
using namespace std;  
int main(){  
 int n, temp;  
   
 cout << "Enter the number: ";  
 cin >> n;  
 temp = n;  
 int sum = 0, count = 0;  
  
 while (temp > 0) {  
 temp = temp / 10;  
 count++;  
 }  
  
 for (int i = 1; i <= count; i++) {  
 int rem = n % 10;  
 sum = sum + rem;  
 n = n / 10;  
 }  
  
 cout << "\nSum of digits of the number: " << sum;  
 return 0;  
}

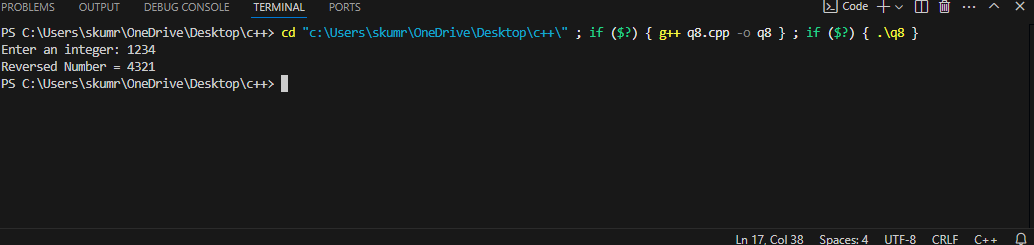
Output for Sum of Digits Program:



**8. Write a program to reverse a given integer.**

#include <iostream>  
using namespace std;  
  
int main() {  
 int n, revnum = 0, remainder;  
  
 cout << "Enter an integer: ";  
 cin >> n;  
  
 while(n != 0) {  
 remainder = n % 10;  
 revnum = revnum \* 10 + remainder;  
 n /= 10;  
 }  
  
 cout << "Reversed Number = " << revnum;  
  
 return 0;  
}

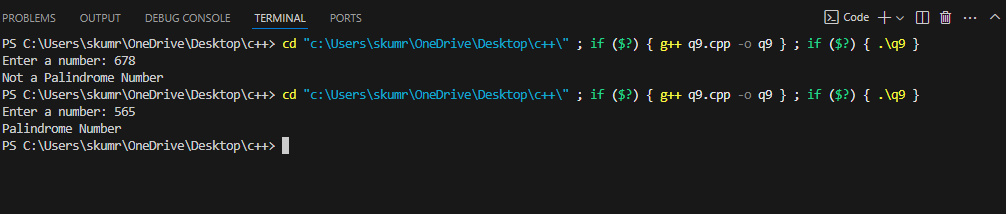
Output for Reverse Number Program:



**9. Write a program to check if a number is a palindrome.**

#include <iostream>  
using namespace std;  
  
int main() {  
 int num, reversed = 0, remainder, original;  
 cout << "Enter a number: ";  
 cin >> num;  
 original = num;  
  
 while (num != 0) {  
 remainder = num % 10;  
 reversed = reversed \* 10 + remainder;  
 num /= 10;  
 }  
  
 if (original == reversed)  
 cout << "Palindrome Number" << endl;  
 else  
 cout << "Not a Palindrome Number" << endl;  
  
 return 0;  
}

Output for Palindrome Number Program:



**10.Write a program to print fibonnaci series upto n terms**

#include <iostream>

using namespace std;

int main() {

    int n, first = 0, second = 1, next;

    cout << "Enter the number of terms: ";

    cin >> n;

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

        cout << first << " ";

        next = first + second;

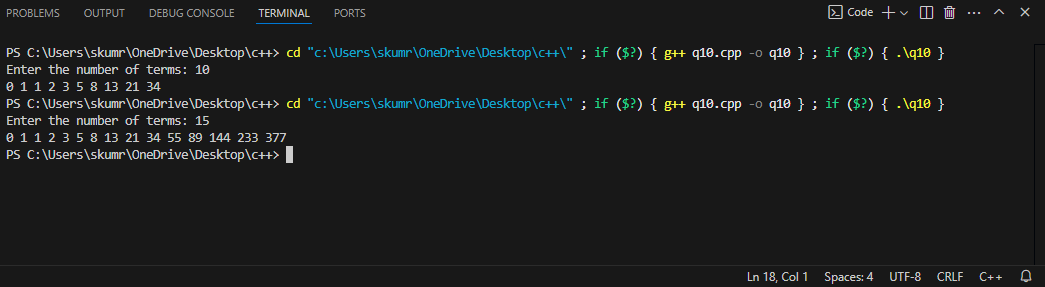
        first = second;

        second = next;

    }

return 0;

}



**11.Write a program to determine given n digit no is Armstrong no or not**

#include<iostream>

#include<cmath>

using namespace std;

int main()

{

    int n, temp, count = 0, rem;

    double sum = 0;

cout << "Enter a number" << endl;

    cin >> n;

  temp = n;

while (n != 0)

    {

count++;

        n /= 10;

    }

n = temp;

    while (n != 0)

    {

        rem = n % 10;

        sum += pow(rem, count);

        n /= 10;

    }

    if (round(sum) == temp)

    {

        cout << temp << " is an Armstrong Number..." << endl;

    }

    else

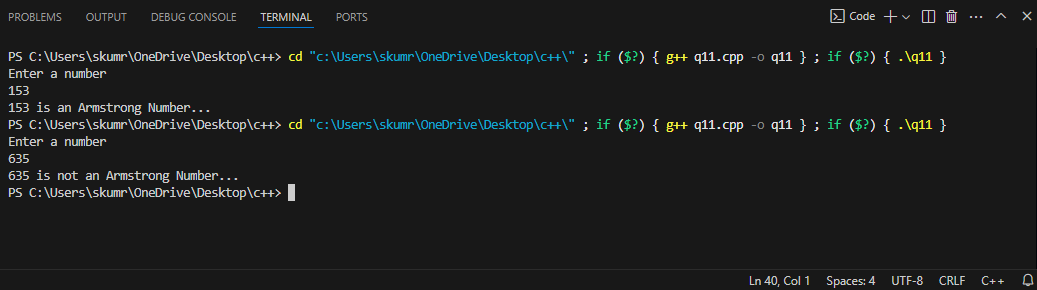
    {

        cout << temp << " is not an Armstrong Number..." << endl;

    }

return 0;

}



**12.Write a program to print all even no.s between 100 and 200**

#include <iostream>

Using namespace std;

int main() {

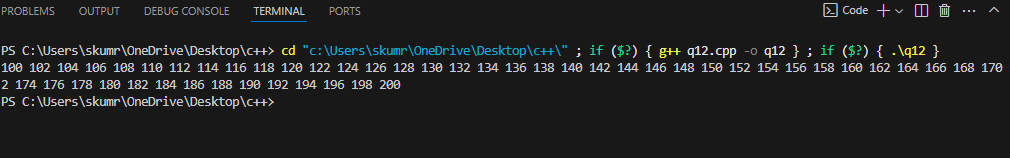
    for (int i = 100; i <= 200; i += 2) {

        cout << i << " ";

    }

    return 0;

}



**13.write a program to print first 50 prime numbers.**

#include <iostream>

using namespace std;

bool isPrime(int num) {

    if (num < 2)

        return false;

    for (int i = 2; i \* i <= num; i++) {

        if (num % i == 0)

          return false;

    }

    return true;

}

int main() {

    int count = 0, num = 2;

    while (count < 50) {

        if (isPrime(num)) {

            cout << num << " ";

            count++;

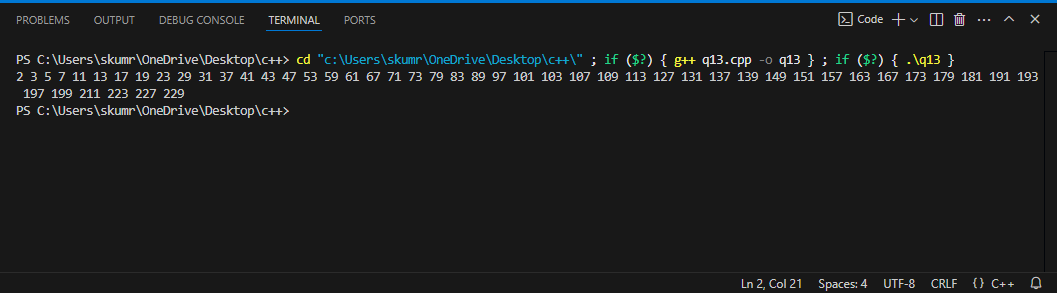
        }

        num++;

    }

    return 0;

}



**14.write a program to print all 4 digit Armstrong No.s**

#include <iostream>

#include <cmath>

using namespace std;

int main() {

    cout << "Four-digit Armstrong numbers are:\n";

    for (int num = 1000; num <= 9999; num++) {

        int originalNum = num;

        int sum = 0;

        int temp = num;

        while (temp > 0) {

            int digit = temp % 10;

            sum += pow(digit, 4); // Raising to power 4

            temp /= 10;

        }

        if (sum == originalNum) {

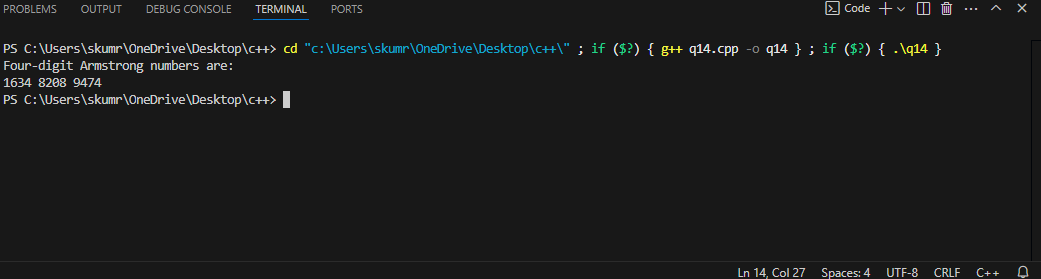
            cout << num << " ";

        }

    }

    return 0;

}



**15. Write a program to print following patterns**

**I**]. #include <iostream>

using namespace std;

int main() {

    int n = 5;

    for (int i = 1; i <= n; i++) {

        for (int j = 1; j <= i; j++)

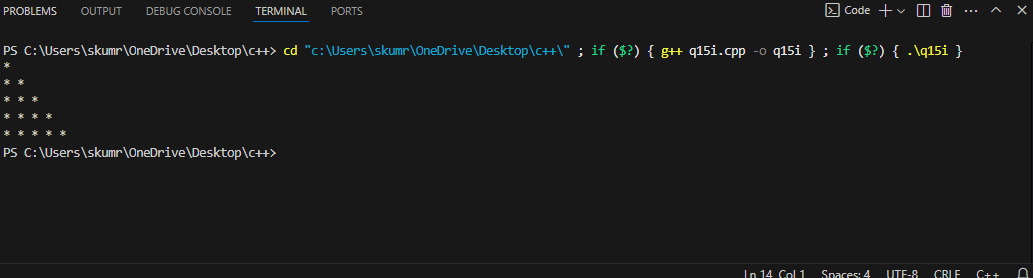
            cout << "\* ";

        cout << endl;

    }

    return 0;

}



II].

#include <iostream>

using namespace std;

int main() {

    int n = 5;

    for (int i = 1; i <= n; i++) {

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

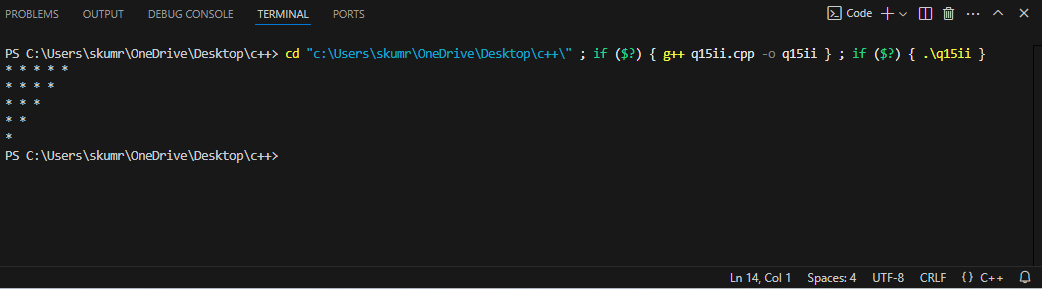
            cout << "\* ";

        cout << endl;

    }

    return 0;

}



IV].

#include <iostream>

using namespace std;

int main() {

    int n = 5;

    int c = 1;

    for (int i = 0; i < n; i++) {

        for (int j = 0; j <= i; j++) {

            cout<<c++;

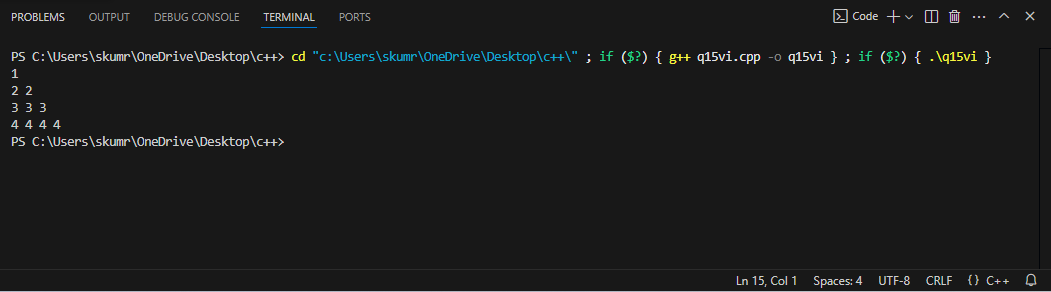
        }

        cout<<"\n";

    }

    return 0;

}



V].

#include <iostream>

using namespace std;

int factorial(int num) {

    int fact = 1;

    for (int i = 1; i <= num; i++)

        fact \*= i;

    return fact;

}

int combination(int n, int r) {

    return factorial(n) / (factorial(r) \* factorial(n - r));

}

int main() {

    int n = 5;

    for (int i = 0; i < n; i++) {

        for (int space = 0; space < n - i - 1; space++)

            cout << "  ";

        for (int j = 0; j <= i; j++)

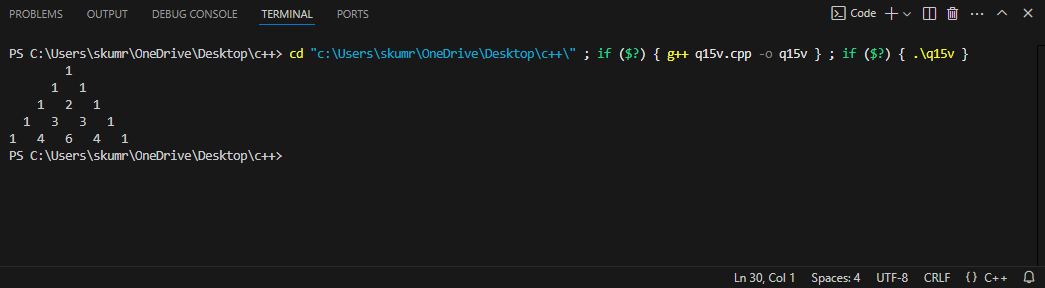
            cout << combination(i, j) << "   ";

        cout << endl;

    }

    return 0;

}



VI].

#include <iostream>

using namespace std;

int main() {

    int rows, columns, n = 4;

    for (rows = 1; rows <= n; rows++) {

        for (columns = 1; columns <= rows; columns++) {

            cout << rows << " ";

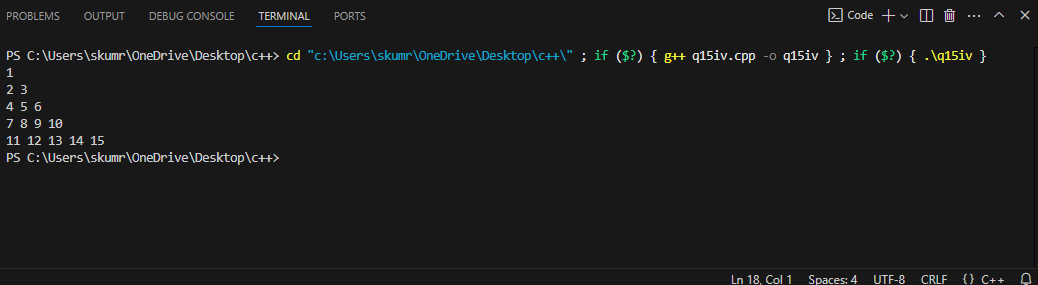
        }

        cout << "\n";

    }

    return 0;

}



**16. Using function write following c++ program**

**I]To print all palindrome numbers for a range 500-1000**

#include <iostream>

using namespace std;

bool isPalindrome(int num) {

    int originalNum = num, reversedNum = 0, digit;

    while (num > 0) {

        digit = num % 10;

        reversedNum = reversedNum \* 10 + digit;

        num /= 10;

    }

    return originalNum == reversedNum;

}

int main() {

    cout << "Palindromes between 500 and 1000:\n";

    for (int i = 500; i <= 1000; i++) {

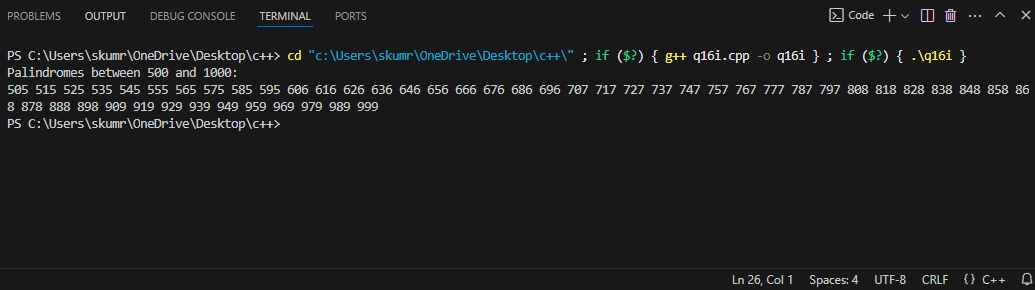
        if (isPalindrome(i))

            cout << i << " ";

    }

    return 0;

}



II]. To print first 100 odd number.

#include <iostream>

using namespace std;

void printOddNumbers(int count) {

    int num = 1;

    for (int i = 0; i < count; i++) {

        cout << num << " ";

        num += 2;

    }

}

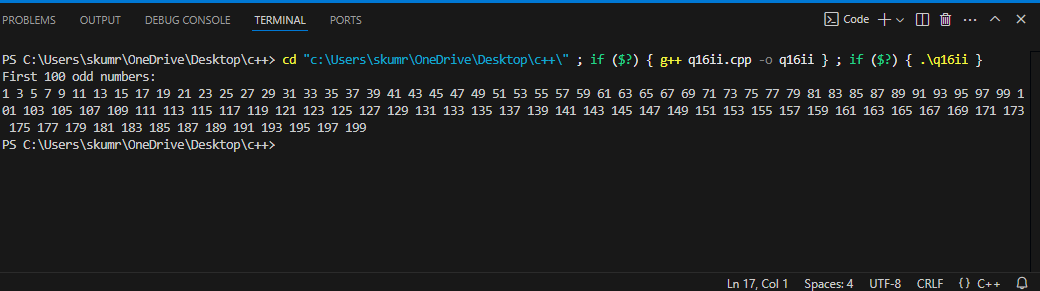
int main() {

    cout << "First 100 odd numbers:\n";

    printOddNumbers(100);

    return 0;

}



**18. Write a program to calculate average of all elements of 1D array.**

#include <iostream>

using namespace std;

int main()

{

    int n, i;

    float num[100], sum=0.0, average;

    cout << "Enter the numbers of data: ";

    cin >> n;

    while (n > 100 || n <= 0)

    {

        cout << "Error! number should in range of (1 to 100)." << endl;

        cout << "Enter the number again: ";

        cin >> n;

    }

    for(i = 0; i < n; ++i)

    {

        cout << i + 1 << ". Enter number: ";

        cin >> num[i];

        sum += num[i];

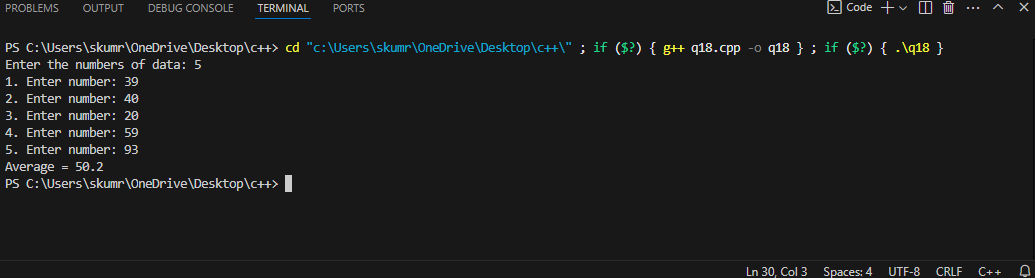
    }

    average = sum / n;

    cout << "Average = " << average;

    return 0;

}



**19. Write a program to find out maximum and minimum value of a 1D numeric array.**

#include <iostream>

using namespace std;

int main(){

    int n,ar[100];

    cout<<"enter the length of array";

    cin>>n;

    for(int i=0;i<n;i++){

        cout << i + 1 << ". Enter number: ";

        cin >> ar[i];

    }

    int max=ar[0],min=ar[0];

    for(int i=0;i<n;i++){

        if(ar[i]<min){

            min=ar[i];

        }

        if(ar[i]>max){

            max=ar[i];

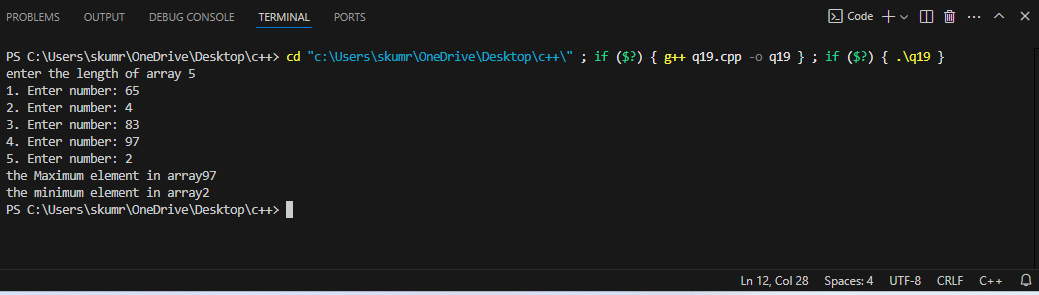
        }

    }

    cout<<"the Maximum element in array"<<max<<endl;

    cout<<"the minimum element in array"<<min;

}



**20. Write a program to find transpose of 2D array.**

#include <iostream>

using namespace std;

int main() {

    int rows, cols;

    cout << "Enter the number of rows and columns: ";

    cin >> rows >> cols;

    int arr[rows][cols], transpose[cols][rows];

    cout << "Enter elements of the matrix:\n";

    for (int i = 0; i < rows; i++) {

        for (int j = 0; j < cols; j++) {

            cin >> arr[i][j];

        }

    }

    for (int i = 0; i < rows; i++) {

        for (int j = 0; j < cols; j++) {

            transpose[j][i] = arr[i][j];

        }

    }

    cout << "Transpose of the matrix:\n";

    for (int i = 0; i < cols; i++) {

        for (int j = 0; j < rows; j++) {

            cout << transpose[i][j] << " ";

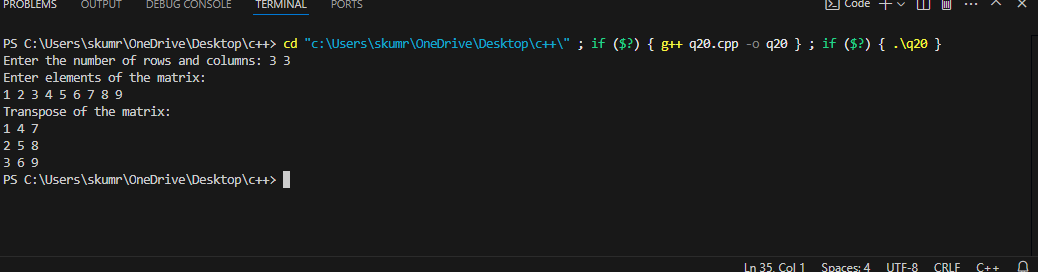
        }

        cout << endl;

    }

    return 0;

}

****

**21.Write a program to add 2D matrices.**

#include <iostream>

using namespace std;

int main() {

    int r, c;

    cout << "Enter number of rows and columns: ";

    cin >> r >> c;

    int A[10][10], B[10][10], result[10][10];

cout << "Enter elements of first matrix:\n";

    for (int i = 0; i < r; i++)

        for (int j = 0; j < c; j++)

            cin >> A[i][j];

cout << "Enter elements of second matrix:\n";

  for (int i = 0; i < r; i++)

        for (int j = 0; j < c; j++)

            cin >> B[i][j];

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

        for (int j = 0; j < c; j++)

            result[i][j] = A[i][j] + B[i][j];

cout << "Resultant Matrix:\n";

    for (int i = 0; i < r; i++) {

        for (int j = 0; j < c; j++)

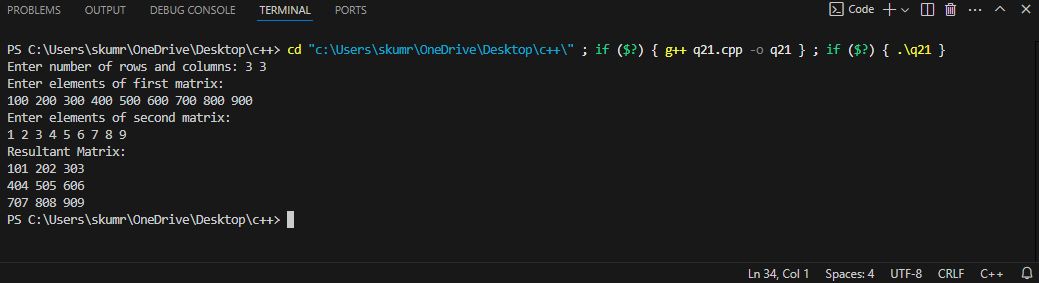
            cout << result[i][j] << " ";

        cout << endl;

    }

    return 0;

}



**22.Write a program to multiply 2D matrices**.

#include <iostream>

using namespace std;

int main() {

    int r1, c1, r2, c2;

    cout << "Enter rows and columns for first matrix: ";

    cin >> r1 >> c1;

    cout << "Enter rows and columns for second matrix: ";

    cin >> r2 >> c2;

    if (c1 != r2) {

        cout << "Matrix multiplication not possible!" << endl;

        return 0;

    }

    int A[10][10], B[10][10], result[10][10] = {0};

    cout << "Enter elements of first matrix:\n";

    for (int i = 0; i < r1; i++)

        for (int j = 0; j < c1; j++)

            cin >> A[i][j];

    cout << "Enter elements of second matrix:\n";

    for (int i = 0; i < r2; i++)

        for (int j = 0; j < c2; j++)

            cin >> B[i][j];

    for (int i = 0; i < r1; i++)

        for (int j = 0; j < c2; j++)

            for (int k = 0; k < c1; k++)

                result[i][j] += A[i][k] \* B[k][j];

    cout << "Resultant Matrix:\n";

    for (int i = 0; i < r1; i++) {

        for (int j = 0; j < c2; j++)

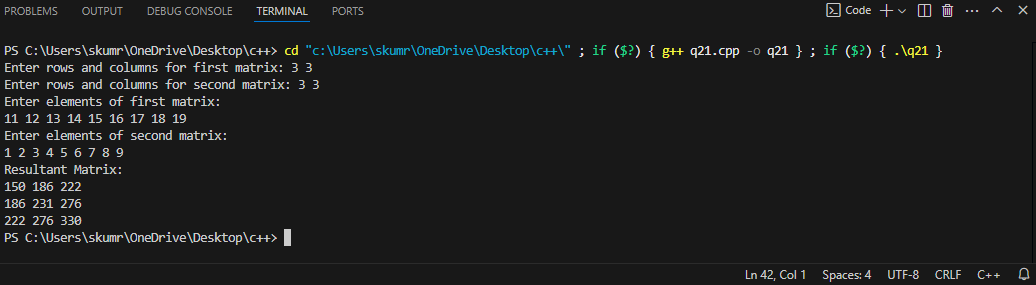
            cout << result[i][j] << " ";

        cout << endl;

    }

    return 0;

}



**23.write a program to sort an array in ascending order.**

#include <iostream>

using namespace std;

void bubbleSort(int arr[], int n) {

for (int i = 0; i < n - 1; i++) {

for (int j = 0; j < n - i - 1; j++) {

if (arr[j] > arr[j + 1]) {

int temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

}

}

void displayArray(int arr[], int n) {

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

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

}

cout << endl;

}

int main() {

int n;

cout << "Enter the number of elements: ";

cin >> n;

int arr[n];

cout << "Enter the elements:" << endl;

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

cin >> arr[i];

}

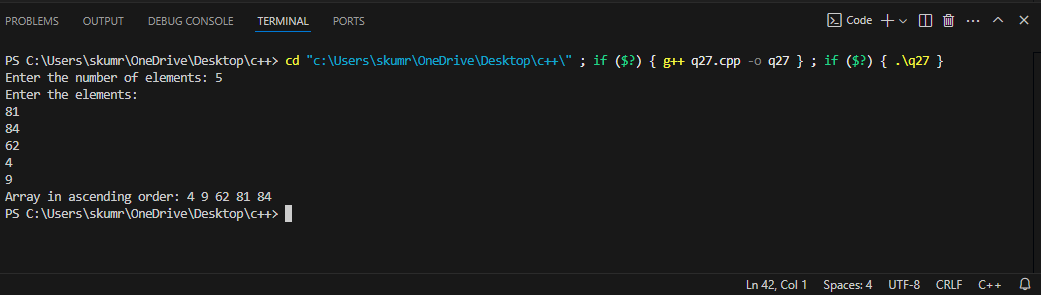
bubbleSort(arr, n);

cout << "Array in ascending order: ";

displayArray(arr, n);

return 0;

}

****

**24.Write a program to reverse a given string.**

#include <iostream>

#include <string>

using namespace std;

int main() {

    string str;

    cout << "Enter a string: ";

    getline(cin, str);

    int len = str.length();

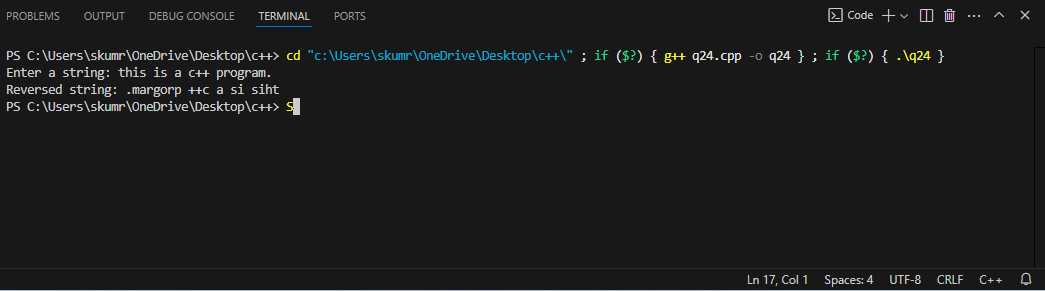
    for (int i = 0; i < len / 2; i++)

        swap(str[i], str[len - i - 1]);

    cout << "Reversed string: " << str << endl;

    return 0;

}



**25.Write a program to count all vowels in a given string.**

#include <iostream>

#include <string>

using namespace std;

int main() {

    string str;

    int count = 0;

    cout << "Enter a string: ";

    getline(cin, str);

    for (char ch : str) {

        ch = tolower(ch);

        if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u')

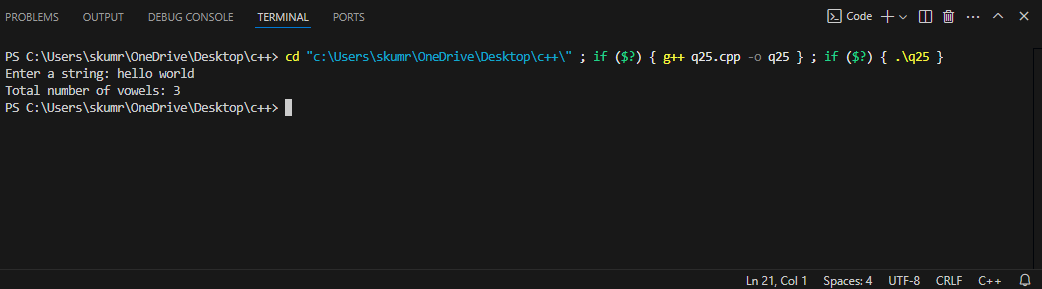
            count++;

    }

    cout << "Total number of vowels: " << count << endl;

    return 0;

}



**26.Write a program to check if a string is palindrome or not.**

#include <iostream>

#include <string>

using namespace std;

int main() {

    string str;

    cout << "Enter a string: ";

    cin >> str;

    int len = str.length();

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

        if (str[i] != str[len - i - 1]) {

            cout << "Not a palindrome." << endl;

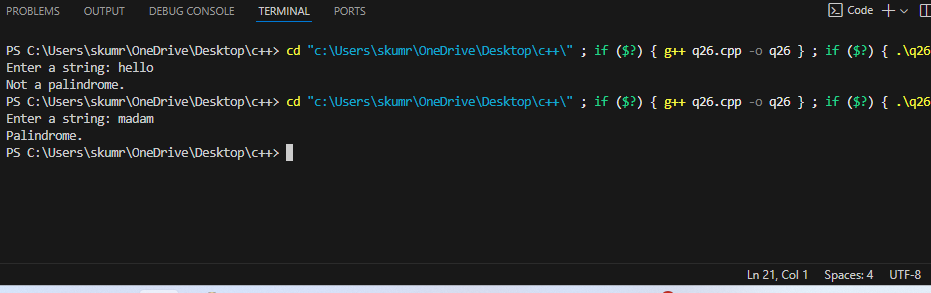
            return 0;

        }

    }

    cout << "Palindrome." << endl;

    return 0;

}

**27.Write a program to check if a given string is anagram not.**

#include <iostream>

#include <algorithm>

using namespace std;

int main() {

    string str1, str2;

    cout << "Enter first string: ";

    cin >> str1;

    cout << "Enter second string: ";

    cin >> str2;

    if (str1.length() != str2.length()) {

        cout << "Not an anagram." << endl;

        return 0;

    }

    sort(str1.begin(), str1.end());

    sort(str2.begin(), str2.end());

    if (str1 == str2)

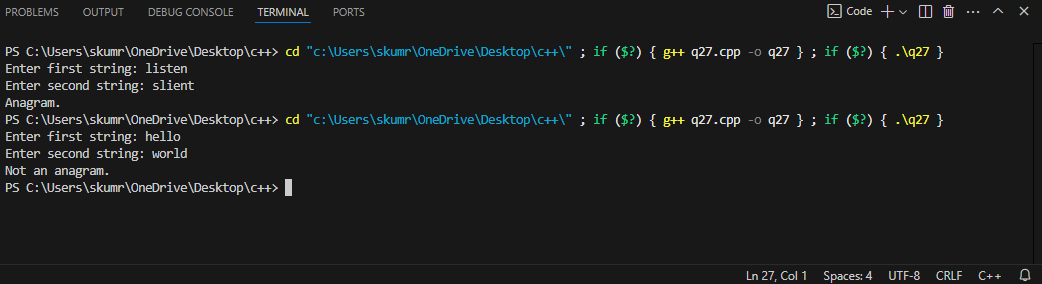
        cout << "Anagram." << endl;

    else

        cout << "Not an anagram." << endl;

    return 0;

}



**28. Write a program to display the minimum, maximum, sum, search and average of elements of an array**.

#include <iostream>

using namespace std;

void arrayOperations(int arr[], int size, int searchElement) {

if (size == 0) {

cout << "The array is empty." << endl;

return;

}

int minElement = arr[0], maxElement = arr[0];

int totalSum = 0;

for (int i = 0; i < size; i++) {

if (arr[i] < minElement) {

minElement = arr[i];

}

if (arr[i] > maxElement) {

maxElement = arr[i];

}

totalSum += arr[i];

}

double average = static\_cast<double>(totalSum) / size;

cout << "Minimum element: " << minElement << endl;

cout << "Maximum element: " << maxElement << endl;

cout << "Sum of elements: " << totalSum << endl;

cout << "Average of elements: " << average << endl;

bool found = false;

for (int i = 0; i < size; i++) {

if (arr[i] == searchElement) {

found = true;

break;

}

}

if (found) {

cout << "Element " << searchElement << " found in the array." << endl;

} else {

cout << "Element " << searchElement << " not found in the array." << endl;

}

}

int main() {

int arr[] = {10, 20, 30, 40, 50};

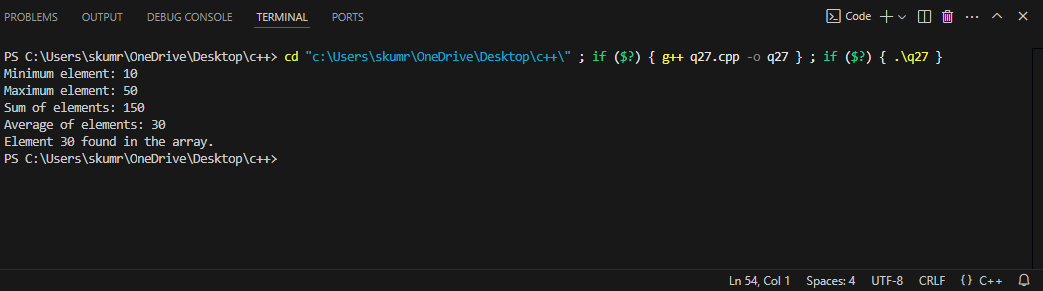
int size = sizeof(arr) / sizeof(arr[0]);

int searchElement = 30;

arrayOperations(arr, size, searchElement);

return 0;

}



**29. Define a class student with the following specification  
Private members of class student  
admno                        integer  
sname                        20 character  
eng. math, science       float  
total                            float  
Public member function of class student**

**ctotal()                        a function to calculate eng + math + science with float return type.  
Takedata()                   Function to accept values for admno, sname, eng, science Showdata()                   Function to display all the data members on the screen**

#include <iostream>

#include <cstring>

using namespace std;

class Student {

private:

    int admno;

    char sname[20];

    float eng, math, science, total;

public:

    void Takedata() {

        cout << "Enter Admission Number: ";

        cin >> admno;

        cout << "Enter Student Name: ";

        cin.ignore();  // To ignore the newline character from previous input

        cin.getline(sname, 20);

        cout << "Enter marks in English: ";

        cin >> eng;

        cout << "Enter marks in Math: ";

        cin >> math;

        cout << "Enter marks in Science: ";

        cin >> science;

    }

    float ctotal() {

        total = eng + math + science;

        return total;

    }

    void Showdata() {

        cout << "Admission Number: " << admno << endl;

        cout << "Student Name: " << sname << endl;

        cout << "Marks in English: " << eng << endl;

        cout << "Marks in Math: " << math << endl;

        cout << "Marks in Science: " << science << endl;

        cout << "Total Marks: " << ctotal() << endl;

    }

};

int main() {

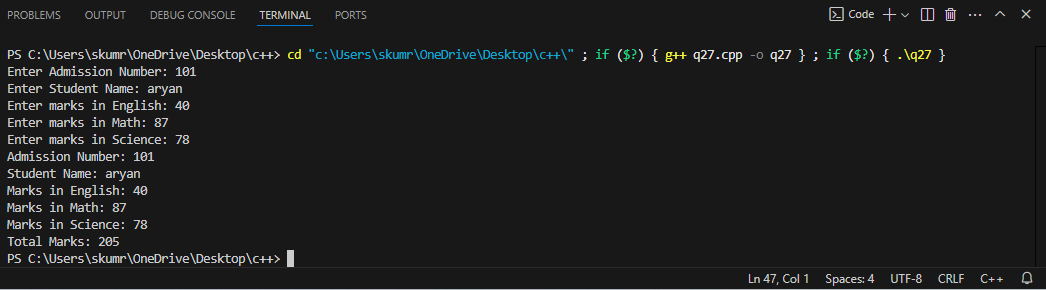
    Student s1;

    s1.Takedata();

    s1.Showdata();

    return 0;

}

****

**30. Define a class in C++ with following description:**

**Private Members  
A data member Flight number of type integer  
A data member Destination of type string  
A data member Distance of type float  
A data member Fuel of type float  
A member function CALFUEL() to calculate the value of Fuel as per the following criteria  
            Distance                                                          Fuel  
            <=1000                                                           500  
            more than 1000  and <=2000                          1100  
            more than 2000                                              2200**

**Public Members**

**A function FEEDINFO() to allow user to enter values for Flight Number, Destination, Distance & call function CALFUEL() to calculate the quantity of Fuel.  
A function SHOWINFO() to allow user to view the content of all the data members.**

#include <iostream>

#include <string>

using namespace std;

class Flight {

private:

    int flightNumber;

    string destination;

    float distance;

    float fuel;

public:

    void CALFUEL() {

        if (distance <= 1000) {

            fuel = 500;

        } else if (distance > 1000 && distance <= 2000) {

            fuel = 1100;

        } else {

            fuel = 2200;

        }

    }

    void FEEDINFO() {

        cout << "Enter Flight Number: ";

        cin >> flightNumber;

        cin.ignore();  // To ignore the newline character after reading the integer

        cout << "Enter Destination: ";

        getline(cin, destination);

        cout << "Enter Distance: ";

        cin >> distance;

        CALFUEL();  // Call CALFUEL() to calculate the fuel

    }

    void SHOWINFO() {

        cout << "Flight Number: " << flightNumber << endl;

        cout << "Destination: " << destination << endl;

        cout << "Distance: " << distance << endl;

        cout << "Fuel Required: " << fuel << " liters" << endl;

    }

};

int main() {

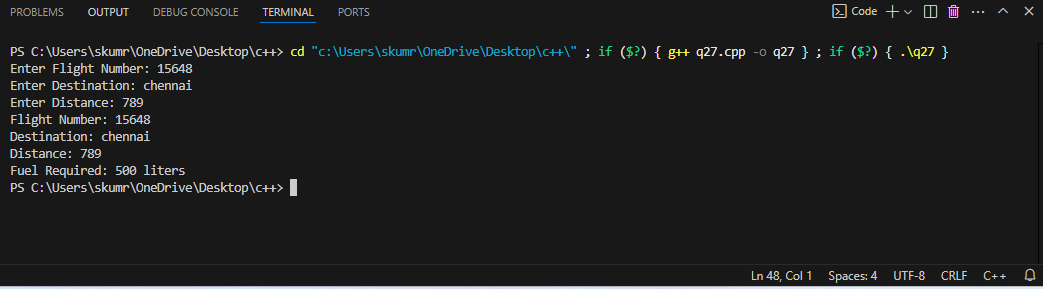
    Flight f1;

    f1.FEEDINFO();  // Input flight details and calculate fuel

    f1.SHOWINFO();  // Display the flight details

    return 0;

}

****

**31. Write a menu driven program to perform following:**

a) Input a matrix

b) Display matrix

c) Add two matrix

d) Multiply two matrixes

e) Transpose a matrix

#include <iostream>

using namespace std;

void inputMatrix(int matrix[10][10], int rows, int cols) {

cout << "Enter elements of the matrix:" << endl;

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

cin >> matrix[i][j];

}

}

}

void displayMatrix(int matrix[10][10], int rows, int cols) {

cout << "Matrix:" << endl;

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

cout << matrix[i][j] << " ";

}

cout << endl;

}

}

void addMatrices(int matrix1[10][10], int matrix2[10][10], int result[10][10], int rows, int cols) {

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

result[i][j] = matrix1[i][j] + matrix2[i][j];

}

}

}

void multiplyMatrices(int matrix1[10][10], int matrix2[10][10], int result[10][10], int rows1, int cols1, int rows2, int cols2) {

if (cols1 != rows2) {

cout << "Matrix multiplication not possible." << endl;

return;

}

for (int i = 0; i < rows1; i++) {

for (int j = 0; j < cols2; j++) {

result[i][j] = 0;

for (int k = 0; k < cols1; k++) {

result[i][j] += matrix1[i][k] \* matrix2[k][j];

}

}

}

}

void transposeMatrix(int matrix[10][10], int result[10][10], int rows, int cols) {

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

result[j][i] = matrix[i][j];

}

}

}

int main() {

int choice, rows1, cols1, rows2, cols2;

int matrix1[10][10], matrix2[10][10], result[10][10];

while (true) {

cout << "\nMenu:" << endl;

cout << "1. Input Matrix" << endl;

cout << "2. Display Matrix" << endl;

cout << "3. Add Two Matrices" << endl;

cout << "4. Multiply Two Matrices" << endl;

cout << "5. Transpose a Matrix" << endl;

cout << "6. Exit" << endl;

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case 1:

cout << "Enter rows and columns for matrix: ";

cin >> rows1 >> cols1;

inputMatrix(matrix1, rows1, cols1);

break;

case 2:

cout << "Enter rows and columns for matrix: ";

cin >> rows1 >> cols1;

displayMatrix(matrix1, rows1, cols1);

break;

case 3:

cout << "Enter rows and columns for matrix 1: ";

cin >> rows1 >> cols1;

inputMatrix(matrix1, rows1, cols1);

cout << "Enter rows and columns for matrix 2: ";

cin >> rows2 >> cols2;

inputMatrix(matrix2, rows2, cols2);

if (rows1 == rows2 && cols1 == cols2) {

addMatrices(matrix1, matrix2, result, rows1, cols1);

displayMatrix(result, rows1, cols1);

} else {

cout << "Matrices cannot be added due to different dimensions." << endl;

}

break;

case 4:

cout << "Enter rows and columns for matrix 1: ";

cin >> rows1 >> cols1;

inputMatrix(matrix1, rows1, cols1);

cout << "Enter rows and columns for matrix 2: ";

cin >> rows2 >> cols2;

inputMatrix(matrix2, rows2, cols2);

multiplyMatrices(matrix1, matrix2, result, rows1, cols1, rows2, cols2);

displayMatrix(result, rows1, cols2);

break;

case 5:

cout << "Enter rows and columns for matrix: ";

cin >> rows1 >> cols1;

inputMatrix(matrix1, rows1, cols1);

transposeMatrix(matrix1, result, rows1, cols1);

displayMatrix(result, cols1, rows1);

break;

case 6:

return 0;

default:

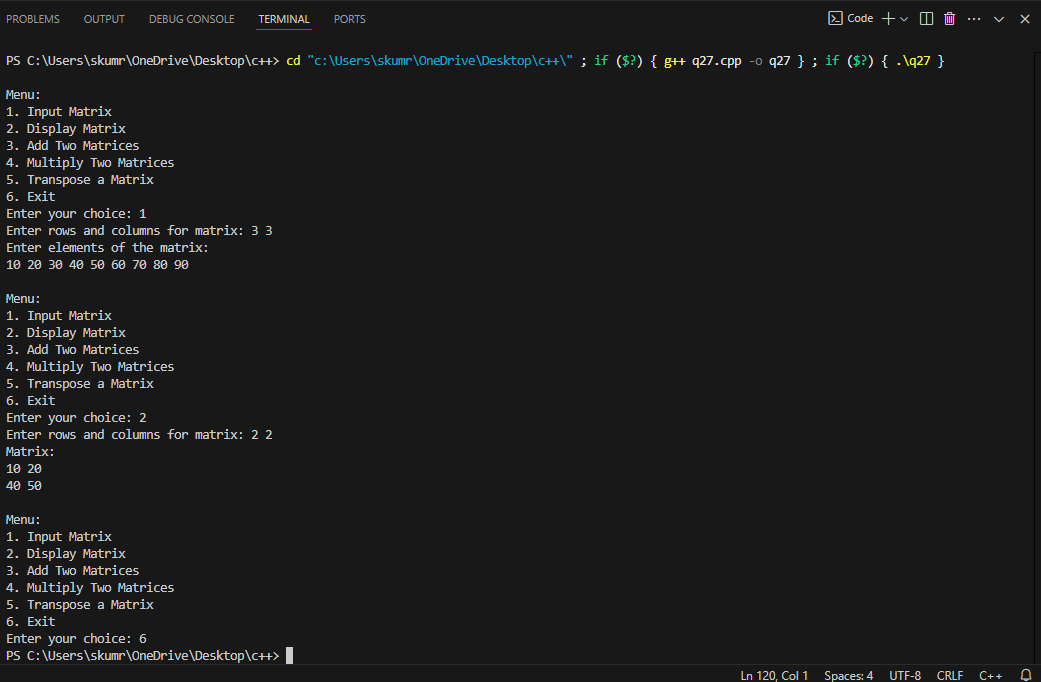
cout << "Invalid choice! Please try again." << endl;

}

}

return 0;

}



**31.Define a class called Car with attributes such as make, model, and year. Include member functions to set and get these attributes. Create an object of the Car class and demonstrate the use of its member functions**

#include <iostream>

#include <string>

using namespace std;

class Car {

private:

string make;

string model;

int year;

public:

void setAttributes(string carMake, string carModel, int carYear) {

make = carMake;

model = carModel;

year = carYear;

}

string getMake() {

return make;

}

string getModel() {

return model;

}

int getYear() {

return year;

}

};

int main() {

Car myCar;

myCar.setAttributes("Toyota", "Corolla", 2020);

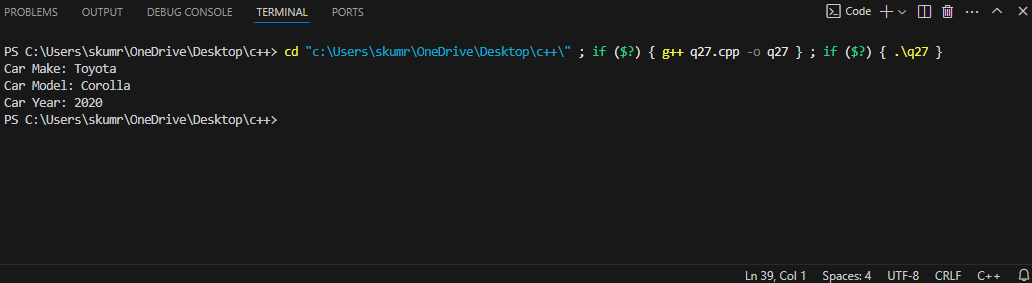
cout << "Car Make: " << myCar.getMake() << endl;

cout << "Car Model: " << myCar.getModel() << endl;

cout << "Car Year: " << myCar.getYear() << endl;

return 0;

}



**32. Define a class called Address with attributes such as street, city, and zipCode. Create a class called Person that has an Address object as a member variable. Demonstrate composition by creating a Person object and accessing its Address attributes.**

#include <iostream>

#include <string>

using namespace std;

class Address {

public:

    string street;

    string city;

    string zipCode;

    void setAddress(string str, string cty, string zip) {

        street = str;

        city = cty;

        zipCode = zip;

    }

    void displayAddress() {

        cout << "Street: " << street << endl;

        cout << "City: " << city << endl;

        cout << "Zip Code: " << zipCode << endl;

    }

};

class Person {

public:

    string name;

    int age;

    Address address;

    void setPersonInfo(string personName, int personAge, string str, string cty, string zip) {

        name = personName;

        age = personAge;

        address.setAddress(str, cty, zip);

    }

    void displayPersonInfo() {

        cout << "Name: " << name << endl;

        cout << "Age: " << age << endl;

        address.displayAddress();

    }

};

int main() {

    Person person1;

    person1.setPersonInfo("John Doe", 30, "123 Main St", "Springfield", "12345");

    person1.displayPersonInfo();

    return 0;

}

