C++ Assignment

LAB QUESTIONS

Q1. WAP to find given number is even or odd.

#include<iostream>

using namespace std;

int main()

{ int n ;

cout << "Enter a number ";

cin >> n;

if (n%2==0)

cout<< "number is Even";

else

cout<< "number is Odd";

return 0;

}

Q2.WAP to find given number is prime or composite.

#include<iostream>

using namespace std;

int main()

{

    int num , factor = 0 ;

    cout << "enter a positive number :";

    cin >> num ;

    if(num == 1 || num < 0)

    cout << "Not a valid choice"<<endl;

    for (int i = 2; i\*i<num;i++)

    { if (num%i==0)

        {factor = 1;

            break;}}

    if (factor == 0)

    cout<< "this is a prime number " << endl;

    else

    cout<< "this is a composite number"<<endl;

    return 0;}

Q3.WAP to print table of a given number upto n multiple.

#include<iostream>

using namespace std;

int main()

{ int n , i;

    cout<< "Enter your number to print its table "<<endl;

    cin>> n;

    cout << "Enter the number of multiples you want in "<< n<< "table"<< endl;

    cin>> i;

    cout<< "Multiplication table is :"<<endl;

    for (int j = 1; j<=i;j++)

    {

      cout<< n << "x"<< j << "="<< n\*j<< endl;

    }

    return 0;

}

Q4.WAP to (i)greater of two number, (ii)greater of three number.

#include<iostream>

using namespace std;

int main()

{ // for two numbers

  int a , b;

  cout<< "Enter the two numbers to compare :"<<endl;

  cin >> a>> b;

  if(a>b)

  cout << a << " is greatest"<<endl;

  else

  cout << b << " is greatest"<< endl;

  // for three numbers

  cout << "ennter three numbers"<<endl;

  int m , n , o , largest;

  cin>> m>>n>>o;

  largest = (m>n)? ((m>o)? m : o) : ((n>o)? n:o);

  cout << "largest number of all three are :"<< largest<< endl;

  return 0;

}

Q5.WAP to find sum of first n natural numbers .

#include<iostream>

using namespace std;

int main()

{

  int n , sum=0;

  cout << "Enter number upto which you want to print the sum " <<endl;

  cin>> n;

  if (n<1)

  cout<<"not a valid input";

  for (int i = 1; i<=n; i++)

  {

    sum += i;

  }

  cout<< "Your sum upto "<< n << " is " << sum<<endl;

return 0;

}

Q6. WAP to find factorial of given number.

#include<iostream>

using namespace std;

int main(){

    int n , fact=1;

  cout << "Enter number of which you want to find factorial" <<endl;

  cin>> n;

  if (n<1)

  cout<<"not a valid input";

  for (int i = 1; i<=n;i++){

    fact \*= i;}

  cout<< "Your factorial of "<< n << " is " << fact <<endl;

return 0;

}

Q7.WAP to find sum of digits of n digit number.

#include<iostream>

using namespace std;

int main(){

    int num , rem , sum= 0;

    cout << "Enter your number :"<<endl;

    cin>> num;

    int n ;

    n = num;

    while (n!=0)

    {

    rem = n%10;

    sum += rem ;

    n = n /10;

    }

    cout << "Sum of digits of "<<num <<" is "<< sum<<endl;

    return 0;

}

Q8.WAP to find reverse of a number

#include<iostream>

using namespace std;

int main(){

    int num , rev = 0 , rem , n ;

    cout << "Enter your number :"<<endl;

    cin>> num;

    n = num ;

    while (n!=0)

    {

        rem = n %10;

        rev = rev \*10 + rem ;

        n /= 10;

    }

    cout<< "The reverse of " << num << " is "<< rev << endl;

    return 0;

}

Q9.WAP to determine given number is palindrome or not

#include<iostream>

using namespace std;

int main(){

    int num, orign, revn = 0, i;

    cout << "Enter a number: ";

    cin >> num; orign = num;

    while (num > 0) {

        i = num % 10;

        revn = revn \* 10 + i;

        num /= 10;}

      if (orign == revn) {

        cout << orign<< " is a palindrome." << endl;

    } else {

        cout << orign << " is not a palindrome." << endl;

    }  return 0}

Q10.WAP to print fabonacci series upto n terms.

#include<iostream>

using namespace std;

int main(){

    int n, a = 0, b = 1, next;

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

    cin >> n;

    cout << "Fibonacci Series: " << a << ", " << b;

    for (int i = 3; i <= n; ++i) {

        next = a + b;

        cout << ", " << next;

        a = b;

        b = next;}

    cout << endl;

    return 0;}

Q11. WAP to determine given n digit no. is Armstrong or not.

#include<iostream>

using namespace std;

int main(){

    int num, originalNum, remainder, result = 0, n = 0, temp, power;

    cout << "Enter a number: ";

    cin >> num;

    originalNum = num;temp = num;

    while (temp != 0) {

        temp /= 10;

        ++n;}

    while (originalNum != 0) {

        remainder = originalNum % 10;

        power = 1;

        for (int i = 0; i < n; i++) {

            power \*= remainder;}

        result += power;

        originalNum /= 10;}

     if (result == num) {

        cout << num << " is an Armstrong number." << endl;

    } else {

        cout << num << " is not an Armstrong number." << endl;}

    return 0;}

Q12.WAP to print all even numbers between 100 & 200.

#include<iostream>

using namespace std;

int main(){

    int i ;

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

    cout << i << "," ;

    return 0;}

Q13.WAP to print first 50 prime numbers.

#include <iostream>

using namespace std;

bool isPrime(int num) {

if (num <= 1) return false;

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

if (num % i == 0) return false;

}

return true;

}

int main() {

int count = 0;

int num = 2;

cout << "First 50 prime numbers are:\n";

while (count < 50) {

if (isPrime(num)) {

cout << num << " ";

count++;

}

num++;

}

cout << endl;

return 0;

}

Q14.WAP to print all 4 digits Armstrong number.

#include <stdio.h>

#include <cmath>

using namespace std;

bool isArmstrong(int num)

{

int originalNum = num; int sum = 0;

while (num > 0)

{ int digit = num % 10; sum += pow(digit, 4); // Raise each digit to the power of 4

num /= 10;

}

return sum == originalNum; }

int main() {

cout << "4-digit Armstrong numbers are:\n";

for (int i = 1000; i <= 9999; i++)

{ if (isArmstrong(i))

{ cout << i << " "; }

}

cout << endl;

return 0; }

Q15.WAP to print following pattern.

1 . Star Right angled Triangle

#include<iostream>

using namespace std;

int main(){

    int i , j;

    for (i = 1 ; i <=5; i++)

    {

  for ( j = 1; j<=i; j++)

          {  cout<<"\*";}

        cout<<endl;

    }

    return 0; }

2.Inverted right angled triangle.

#include<iostream>

using namespace std;

int main(){

    int i , j;

    for (i = 1 ; i <=5; i++)

    {

        for ( j = 1; j<=6- i; j++)

        {

            cout<<"\*";

        }

        cout<<endl;

    }

    return 0;

}

3. Triangle star pattern

#include<iostream>

using namespace std;

int main(){

    int i , j;

    for (i=1; i<=5;i++)  {

        for (j=1;j<=9;j++) {

            if (j>=(5-i+1)&& j<=(5 + i-1))

            cout<<" \* ";

            else

            cout<<"   ";

        } cout<< endl;

    }return 0;

}

4. Numbered right angled triangle

#include<iostream>

using namespace std;

int main(){

    int i , j;

    for (i = 1 ; i <=4; i++) {

        for ( j = 1; j<=i; j++) {

            cout<< i;

        }  cout<<endl;

    } return 0;

}

5. Pascal’s triangle

#include <iostream>

using namespace std;

long long factorial(int n) {

long long fact = 1;

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

fact \*= i;

return fact;

}

void printPascalsTriangle(int rows) {

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

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

cout << " ";

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

cout << factorial(i) / (factorial(j) \* factorial(i - j)) << " ";

} cout << endl;

}

}

int main() {

int n;

cout << "Enter number of rows: ";

cin >> n;

printPascalsTriangle(n);

return 0;

}

6.Flyodd’s triangle

#include <iostream>

using namespace std;

int main() {

int n;

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

cin >> n;

int num=1;

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

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

cout << num << " ";

num++;

} cout << endl;

} return 0;

}

Q16.Using functions , write following c++ programs .

1. To print all palindromes for a range 500 to 1000.

#include <iostream>

using namespace std;

bool isPalindrome(int num) {

int original = num, reversed = 0, digit;

while (num > 0) {

digit = num % 10;

reversed = reversed \* 10 + digit;

num /= 10;

} return original == reversed;

}

void printPalindromes() {

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

if (isPalindrome(i)) {

cout << i << " ";

}

} cout << endl;

}

int main() {

printPalindromes();

return 0;}

2. To print first 100 odd numbers

#include <iostream>

using namespace std;

void printOddNumbers() {

    for (int i = 1; i < 200; i += 2) {

        cout << i << " ";}

    cout << endl;

}

int main() {

    printOddNumbers();

    return 0;

}

3.To find binary, octal , hexadecimal equivalent of a given decimal number.

#include <iostream>

using namespace std;

void decimalToBinary(int num) {

int binary[32], i = 0;

while (num > 0) {

binary[i] = num % 2;

num /= 2;

i++;

}

cout << "Binary: ";

for (int j = i - 1; j >= 0; j--) {

cout << binary[j];

}

cout << endl;

}

void decimalToOctal(int num) {

int octal[32], i = 0;

while (num > 0) {

octal[i] = num % 8;

num /= 8;

i++;

}

cout << "Octal: ";

for (int j = i - 1; j >= 0; j--) {

cout << octal[j];

}

cout << endl;

}

void decimalToHexadecimal(int num) {

char hex[32];

int i = 0;

while (num > 0) {

int remainder = num % 16;

if (remainder < 10)

hex[i] = remainder + '0';

else

hex[i] = remainder - 10 + 'A';

num /= 16;

i++;

}

cout << "Hexadecimal: ";

for (int j = i - 1; j >= 0; j--) {

cout << hex[j];

}

cout << endl;

}

int main() {

int num;

cout << "Enter a decimal number: ";

cin >> num;

decimalToBinary(num);

decimalToOctal(num);

decimalToHexadecimal(num);

return 0;

}

4. To calculate geometric sum upto n terms.

#include <iostream>

#include <cmath>

using namespace std;

double geometricSum(int n, double r) {

double sum = 0;

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

sum += pow(r, i);

}

return sum;

}

int main() {

int n;

double r;

cout << "Enter the number of terms (n): ";

cin >> n;

cout << "Enter the common ratio (r): ";

cin >> r;

double result = geometricSum(n, r);

cout << "Geometric Sum up to " << n << " terms: " << result << endl;

return 0;

}

Q17.Using recursion , write a c++ program to :

1.print binary number for a decimal number.

#include <iostream>

using namespace std;

void decimalToBinary(int num) {

if (num == 0) {

return;

}

decimalToBinary(num / 2);

cout << num % 2;

}

int main() {

int num;

cout << "Enter a decimal number: ";

cin >> num;

if (num == 0) {

cout << "Binary: 0" << endl;

} else {

cout << "Binary: ";

decimalToBinary(num);

cout << endl;

}

return 0;

}

2.Print octal number for a decimal number .

#include <iostream>

using namespace std;

void decimalToOctal(int num) {

if (num == 0) {

return;

}

decimalToOctal(num / 8);

cout << num % 8;

}

int main() {

int num;

cout << "Enter a decimal number: ";

cin >> num;

if (num == 0) {

cout << "Octal: 0" << endl;

} else {

cout << "Octal: ";

decimalToOctal(num);

cout << endl;

}

return 0;

}

3. Print factorials for a given range

#include <iostream>

using namespace std;

long long factorial(int num) {

if (num == 0 || num == 1) {

return 1;

} return num \* factorial(num - 1);

}

void printFactorials(int start, int end) {

if (start > end) {

return;

}

cout << "Factorial of " << start << " is " << factorial(start) << endl;

printFactorials(start + 1, end);}

int main() {

int start, end;

cout << "Enter the start of the range: ";

cin >> start;

cout << "Enter the end of the range: ";

cin >> end;

if (start > end) {

cout << "Invalid range!" << endl;

} else {

printFactorials(start, end);

} return 0;

}

4.Print first n terms of Fibonacci series

**#include <iostream>**

**using namespace std;**

**int fibonacci(int n) {**

**if (n == 0) return 0;**

**if (n == 1) return 1;**

**return fibonacci(n - 1) + fibonacci(n - 2);**

**}**

**void printFibonacciSeries(int n, int i = 0) {**

**if (i >= n) return;**

**cout << fibonacci(i) << " ";**

**printFibonacciSeries(n, i + 1);**

**}**

**int main() {**

**int n;**

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

**cin >> n;**

**cout << "Fibonacci Series: ";**

**printFibonacciSeries(n);**

**cout << endl;**

**return 0;**

**}**

Q18.WAP to calculate average of all elements of 1d array.

#include<iostream>

using namespace std;

int main(){

    int arr[5] = { 1 , 2 , 3 , 4 , 5};

    int sum = 0;

    cout << "Elements of arrays are " ;

for (int i = 0;i< 5;i++)

    {  cout<< arr[i] <<" " ;

        sum += arr[i];

    }

    cout<< endl<<"Average of elements of array are : " << sum/5;

return 0;

}

Q19.WAP to find out minimum and maximum value of a 1d numeric array.

#include<iostream>

using namespace std;

int main(){

    int n ;

    cout<< "Enter size of an array " << endl ;

    cin>> n ;

    int arr[n];

    cout << "Enter " << n << " elements: ";

    for (int i = 0; i < n; i++) {

        cin >> arr[i]; }

    int max =arr[0] , min = arr[0];

    for (int i = 1; i < n; i++) {

        if (arr[i] > max) {

            max = arr[i];

        }

        if (arr[i] < min) {

            min = arr[i]; }

    } cout << "Maximum element of an array is : " << max << endl << "Minimum element of an array is : " << min << endl;

   return 0;

}

Q20.WAP to find transpose of 2d matrix .

#include<iostream>

using namespace std;

int main(){

    int matrix[3][3], transpose[3][3];

    cout << "Enter the 9 elements of the matrix:" << endl;

    for (int i = 0; i < 3; i++) {

        for (int j = 0; j < 3; j++) {

            cin >> matrix[i][j];}

    }

    for (int i = 0; i < 3; i++) {

        for (int j = 0; j < 3; j++) {

            transpose[j][i] = matrix[i][j];  }

    }

    cout << "\nOriginal Matrix:" << endl;

    for (int i = 0; i < 3; i++) {

        for (int j = 0; j < 3; j++) {

         cout << matrix[i][j] << " ";

        }  cout << endl;

    }

    cout << "\nTranspose Matrix:" << endl;

    for (int i = 0; i < 3; i++) {

        for (int j = 0; j < 3; j++) {

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

        } cout << endl;

    } return 0;

}

Q21.WAP to add 2d matrices.

#include<iostream>

using namespace std;

int main(){

    int m1[3][3], m2[3][3];

    cout << "Enter the 9 elements of the matrix one:" << endl;

    for (int i = 0; i < 3; i++) {

        for (int j = 0; j < 3; j++) {

            cin >> m1[i][j]; }

    }cout << "Enter the 9 elements of the matrix two:" << endl;

    for (int i = 0; i < 3; i++) {

        for (int j = 0; j < 3; j++) {

            cin >> m2[i][j];  }

    }  int sum[3][3];

    for (int i = 0; i < 3; i++) {

        for (int j = 0; j < 3; j++) {

          sum[i][j] =  m1[i][j] + m2[i][j];

        } cout << endl;

    }  cout << "Sum of these two matrices are" << endl;

    for (int i = 0; i < 3; i++) {

        for (int j = 0; j < 3; j++) {

            cout << sum[i][j] << " ";

        }  cout << endl;

    } return 0;}

Q22.WAP to multiply 2d matrices.

#include <iostream>

using namespace std;

int main() {

int r1, c1, r2, c2;

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

cin >> r1 >> c1;

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

cin >> r2 >> c2;

if (c1 != r2) {

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

return 0;

}

int A[r1][c1], B[r2][c2], result[r1][c2] = {0};

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

for (int i = 0; i < r1; i++) {

for (int j = 0; j < c1; j++) {

cin >> A[i][j]; }

}

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

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 after multiplication:" << endl;

for (int i = 0; i < r1; i++) {

for (int j = 0; j < c2; j++) {

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

cout << endl;

}

return 0;

}

Q23. WAP to sort an array in ascending order.

#include <iostream>

using namespace std;

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

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

int minIndex = i;

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

if (arr[j] < arr[minIndex]) {

minIndex = j; }

}

if (minIndex != i) {

int temp = arr[i];

arr[i] = arr[minIndex];

arr[minIndex] = temp;} }

}

void printArray(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 " << n << " elements: ";

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

cin >> arr[i];

}

selectionSort(arr, n);

cout << "Sorted array in ascending order: ";

printArray(arr, n);

return 0;

}

Q24.WAP to reverse a given string .

#include <iostream>

using namespace std;

int main() {

    string s, rev;

    cout << "Enter a string: ";

    getline(cin, s);

    for (int i = s.length() - 1; i >= 0; i--) {

        rev += s[i];

    }cout << "Reversed string: " << rev << endl;

    return 0;

}

Q25.WAP to count all vowels in a given string .

#include <iostream>

#include <cctype>

using namespace std;

int main() {

    string str;

    int vowelCount = 0;

    cout << "Enter a string: ";

    getline(cin, str);

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

        char ch = tolower(str[i]);

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

            vowelCount++; }

    }

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

    return 0;

}

Q26.WAP to check if a given string is palindrome or not.

#include<iostream>

#include<string>

using namespace std;

int main(){

   string str;

   int flag = 1;

   cout << "Enter a string: ";

   getline(cin, str);

   int len = str.length();

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

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

           flag = 0; // Not a palindrome

           break;}

   }

   if (flag == 1)

       cout << "The string is a palindrome.\n";

   else

       cout << "The string is not a palindrome.\n";

 return 0;

}

Q27.WAP to check if a given string is anagram or 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 << "The strings are not anagrams.\n";

        return 0;}

    sort(str1.begin(), str1.end());

    sort(str2.begin(), str2.end());

    if (str1 == str2)

        cout << "The strings are anagrams.\n";

    else

        cout << "The strings are not anagrams.\n";

    return 0;

}

LAB ASSIGNMENT QUESTIONS

Q28. Write a program to display the minimum, maximum, sum, search and average of elements of an array.

#include <iostream>

using namespace std;

int min(int arr[], int size) {

    int min = arr[0];

    for (int i = 1; i < size; i++) {

        if (arr[i] < min) {

            min = arr[i];

        } }

    return min;

}

int max(int arr[], int size) {

    int max = arr[0];

    for (int i = 1; i < size; i++) {

        if (arr[i] > max) {

            max = arr[i]; }

    } return max;

}

double sum(int arr[], int size) {

    double sum = 0.0;

    for (int i = 0; i < size; i++) {

        sum += arr[i];

    } return sum;

}

double average(int arr[], int size) {

    return sum(arr, size) / size;}

int searchElement(int arr[], int size, int key) {

    for (int i = 0; i < size; i++) {

        if (arr[i] == key) {

            return 1;

        } } return 0;

}

int main() {

    int arr[] = {10, 20, 5, 30, 25};

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

    cout << "Minimum element: " << min(arr, size) << endl;

    cout << "Maximum element: " << max(arr, size) << endl;

    cout << "Sum of elements: " << sum(arr, size) << endl;

    cout << "Average of elements: " << average(arr, size) << endl;

    int key;

    cout << "Enter an element to search: ";

    cin >> key;

    cout << key << (searchElement(arr, size, key) ? " is present in the array." : " is not present in the array.") << endl;

    return 0;

}

Q29. 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;

    float total;

    float ctotal() {

        return eng + math + science;}

public:

    void Takedata() {

        cout << "Enter admission number: ";

        cin >> admno;

        cout << "Enter student name: ";

        cin.ignore();// to ignore empty space.

        cin.getline(sname, 20);

        cout << "Enter marks in English, Math, and Science: ";

        cin >> eng >> math >> science;

        total = ctotal();

    } void Showdata() {

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

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

        cout << "Marks - English: " << eng << ", Math: " << math << ", Science: " << science << endl;

        cout << "Total Marks: " << total << endl;

    }};

int main() {

    Student s;

    s.Takedata();

    cout << "\nStudent Details:\n";

    s.Showdata();

    return 0;

}

Q30. 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>

using namespace std;

class Flight {

private:

    int flightNumber;

    string destination;

    float distance;

    float fuel;

    void CALFUEL(float dist) {

        if (dist <= 1000)

            fuel = 500;

        else if (dist > 1000 && dist <= 2000)

            fuel = 1100;

        else

            fuel = 2200; }

public:

    void FEEDINFO() {

        cout << "Enter Flight Number: ";

        cin >> flightNumber;

        cout << "Enter Destination: ";

        cin.ignore();

        getline(cin, destination);

        cout << "Enter Distance: ";

        cin >> distance;

        CALFUEL(distance); }

    void SHOWINFO() {

        cout << "\nDescription of Flight." << endl;

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

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

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

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

    }

};

int main() {

    Flight indigo;

    indigo.FEEDINFO();

    indigo.SHOWINFO();

    return 0;

}

Q31. 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;

const int MAX = 10;

void inputMatrix(int mat[MAX][MAX], int &rows, int &cols) {

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

    cin >> rows >> cols;

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

    for (int i = 0; i < rows; i++) {

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

            cin >> mat[i][j];  }

    }

}

void displayMatrix(int mat[MAX][MAX], int rows, int cols) {

    cout << "Matrix:\n";

    for (int i = 0; i < rows; i++) {

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

            cout << mat[i][j] << " "; }

        cout << endl;  }

}

void addMatrix(int mat1[MAX][MAX], int mat2[MAX][MAX], int res[MAX][MAX], int rows, int cols) {

    for (int i = 0; i < rows; i++) {

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

            res[i][j] = mat1[i][j] + mat2[i][j];  }

    }

}

void multiplyMatrix(int mat1[MAX][MAX], int mat2[MAX][MAX], int res[MAX][MAX], int r1, int c1, int r2, int c2) {

    if (c1 != r2) {

        cout << "Matrix multiplication not possible.\n";

        return;

    }

    for (int i = 0; i < r1; i++) {

        for (int j = 0; j < c2; j++) {

            res[i][j] = 0;

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

                res[i][j] += mat1[i][k] \* mat2[k][j]; }

        }

    }

}

void transposeMatrix(int mat[MAX][MAX], int trans[MAX][MAX], int rows, int cols) {

    for (int i = 0; i < rows; i++) {

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

            trans[j][i] = mat[i][j];  }

    }

}

int main() {

    int mat1[MAX][MAX], mat2[MAX][MAX], result[MAX][MAX];

    int rows1, cols1, rows2, cols2;

    int choice;

    do {

        cout << "\nMenu:\n";

        cout << "1. Input a matrix\n2. Display matrix\n3. Add two matrices\n4. Multiply two matrices\n5. Transpose a matrix\n6. Exit\n";

        cout << "Enter your choice: ";

        cin >> choice;

        switch (choice) {

            case 1:

                inputMatrix(mat1, rows1, cols1);

                break;

            case 2:

                displayMatrix(mat1, rows1, cols1);

                break;

            case 3:

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

                inputMatrix(mat2, rows2, cols2);

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

                    addMatrix(mat1, mat2, result, rows1, cols1);

                    displayMatrix(result, rows1, cols1);

                } else {

                    cout << "Matrix addition not possible.\n";

                }

                break;

            case 4:

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

                inputMatrix(mat2, rows2, cols2);

                if (cols1 == rows2) {

                    multiplyMatrix(mat1, mat2, result, rows1, cols1, rows2, cols2);

                    displayMatrix(result, rows1, cols2);

                } else {

                    cout << "Matrix multiplication not possible.\n";

                }

                break;

            case 5:

                transposeMatrix(mat1, result, rows1, cols1);

                displayMatrix(result, cols1, rows1);

                break;

            case 6:

                cout << "Exiting...\n";

                break;

            default:

                cout << "Invalid choice. Try again.\n";}

    } while (choice != 6);

  return 0;

}

EXTRA QUESTIONS

Q.32. 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>

using namespace std;

class car{

    string make;

    string model;

    int year;

    public :

    void set() {

        cout<<"Enter the Maker of car : "<< endl;

        cin>>make;

        cout<<"Enter the Model of car : "<< endl;

        cin>>model;

        cout<<"Enter the Year of car : "<< endl;

        cin>>year;

    };

    void get()

    {

        cout<<" the Maker of car is : "<< make<< endl;

        cout<<" the Model of car is : "<< model <<endl;

        cout<<" the Year in which car is launched is: "<< year<< endl;

    };

};

int main(){

    car c1;

    c1.set();

    c1.get();

   return 0;

}

Q33. 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;

};

class Person {

public:

    string name;

    Address address;

    void displayInfo() const {

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

        cout << "Address: " << address.street << ", " << address.city << ", " << address.zipCode << endl;

    }

};

int main() {

    Address homeAddress;

  homeAddress.street = "123 Main St";

    homeAddress.city = "New York";

    homeAddress.zipCode = "10001";

    Person person;

    person.name = "John Doe";

    person.address = homeAddress;

    person.displayInfo();

    cout << "City: " << person.address.city << endl;

     return 0;

}