

Q 1. WAP to find the number is even or odd?

```
#include<iostream>

using namespace std;

int main(){

    int n;

    cout<<"enter the number:";

    cin>>n;

    if(n%2==0){

        cout<<"the number is even"<<endl;

    }

    else{

        cout<<"the number is odd"<<endl;

    }

    return 0;

}
```

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

```
#include<iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int n,i,z = 0;
```

```
    cout << "Enter a number: ";
```

```
    cin >> n;
```

```
    if (n <= 1) {
```

```
        cout << n << " is not a prime number.";
```

```
        return 0;
```

```
    }
```

```
    for (i = 2; i < n; i++) {
```

```
        if (n % i == 0) {
```

```
            z = 1;
```

```
            break;
```

```
        }
```

```
    if (z == 0)
```

```
        cout << n << " is a Prime number.";
```

```
    else
```

```
        cout << n << " is a Composite number.";
```

```
    return 0;
```

```
}
```

Q3.WAP to print the table of the given number to n multiple?

```
#include<iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int n,i,num;
```

```
    cout<<"enter the number:";
```

```
    cin>>num;
```

```
    cout<<"how many multiples:";
```

```
    cin>>n;
```

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

```
        int t = num*i;
```

```
        cout<<t<<endl;
```

```
    }
```

```
    return 0;
```

```
}
```

Q4. WAP to find number

(i) greater between two number.

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

int main() {
    int a, b;
    cout << "Enter two numbers: ";
    cin >> a >> b;

    if (a > b)
        cout << a << " is greater than " << b << ".";
    else if (b > a)
        cout << b << " is greater than " << a << ".";
    else
        cout << "Both numbers are equal.";

    return 0;
}
```

(ii) greater between three numbers.

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

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

    cout << "Enter the three numbers:" << endl;
    cin >> a >> b >> c;

    (a > b) ? (a > c ? cout << a << " is greater" : cout << c << " is greater") : (b > c ? cout << b << " is greater" : cout << c << " is greater");

    return 0;
}
```

Q5. Program to find sum of n natural numbers

```
#include<iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int n, sum = 0;
```

```
    cout << "Enter the value of n: ";
```

```
    cin >> n;
```

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

```
        sum += i;
```

```
    }
```

```
    cout << "Sum of first " << n << " natural numbers is: " << sum << endl;
```

```
    return 0;
```

```
}
```

Q6. Wap to find factorial of the given number

```
#include<iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int n, fact=1;
```

```
    cout << "Enter the value of n: ";
```

```
    cin >> n;
```

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

```
        fact = fact*i;
```

```
    }
```

```
    cout << "the factorial is:" << fact << endl;
```

```
    return 0;
```

```
}
```

Q7 wap to find the sum of digit of the given number

```
#include<iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int num,sum=0;
```

```
    cout<<"enter the number:";
```

```
    cin>>num;
```

```
    while(num>0){
```

```
        int r = num%10;
```

```
        num = num/10;
```

```
        sum = sum +r;
```

```
    }
```

```
    cout<<sum;
```

```
    return 0;
```

```
}
```

Q8 program to write the reverse of a number?

```
#include<iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int num,rev=0;
```

```
    cout<<"enter the number:";
```

```
    cin>>num;
```

```
    while(num>0){
```

```
        int r = num%10;
```

```
        num = num/10;
```

```
        rev = (rev*10) + r;
```

```
    }
```

```
    cout<<rev;
```

```
    return 0;
```

```
}
```


Q9 to find the number is palindrome or not

```
#include<iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int num,rev=0;
```

```
    int original;
```

```
    cout<<"enter the number:";
```

```
    cin>>num;
```

```
    original=num;
```

```
    while(num>0){
```

```
        int r = num%10;
```

```
        num = num/10;
```

```
        rev = (rev*10) + r;
```

```
    }
```

```
    if(original==rev){
```

```
        cout<<"the number is palindrome."<<endl;
```

```
    }
```

```
    else{
```

```
        cout<<"the number is not a palindrome."<<endl;
```

```
    }
```

```
    return 0;
```

```
}
```

Q10 print Fibonacci series

```
#include<iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int a ,b ,c;
```

```
    int n;
```

```
    cout<<"enter n:";
```

```
    cin>>n;
```

```
    a=0;
```

```
    b=1;
```

```
    cout<<"0 1 ";
```

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

```
        c=a+b;
```

```
        a=b;
```

```
        b=c;
```

```
        cout<<c<<" ";
```

```
    }
```

```
    return 0;
```

```
}
```

Q11. Wap to find the number is Armstrong or not.

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int num, originalNum, remainder, result = 0;
```

```
    cout << "Enter a three-digit integer: ";
```

```
    cin >> num;
```

```
    originalNum = num;
```

```
    while (originalNum != 0) {
```

```
        remainder = originalNum % 10;
```

```
        result = result + (remainder * remainder * remainder);
```

```
        originalNum /= 10;
```

```
    }
```

```
    if (result == num)
```

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

```
    else
```

```
        cout << num << " is not an Armstrong number.";
```

```
    return 0;
```

```
}
```

Q12. Wap to write even nmbers between 100 to 200.

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

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

```
        if(i%2==0){
```

```
            cout<<i<<endl;
```

```
        }
```

```
    }
```

```
    return 0;
```

```
}
```

Tejasv Patel (IT-2k24-89)

Q13. Wap to find first 50 prime numbers.

```
#include<iostream>
```

```
using namespace std;
```

```
int main (){
```

```
    int n,i,j,count=1;
```

```
    cout<<"first 50 prime no are:"<<endl;
```

```
    for(n=1;n<1000;n++){
```

```
        int a=0;
```

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

```
            if(n%j==0){
```

```
                a++;
```

```
            }
```

```
        }
```

```
        if(a==2){
```

```
            cout<<count<<":"<<n<<endl;
```

```
            count++;
```

```
            if(count>50)
```

```
            {
```

```
                break;
```

```
            }
```

```
        }
```

```
    }
```

```
    return 0;
```

```
}
```

Q14.program to print all 4 digit Armstrong numbers.

```
#include<iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    cout << "4-digit Armstrong numbers are: " << endl;
```

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

```
        int original = num;
```

```
        int sum = 0;
```

```
        int temp = num;
```

```
        while (temp > 0) {
```

```
            int rem = temp % 10;
```

```
            sum = sum + (rem*rem*rem*rem);
```

```
            temp = temp/10;
```

```
        }
```

```
        if (sum == original) {
```

```
            cout << original << endl;
```

```
        }
```

```
    }
```

```
    return 0;
```

```
}
```

Q15. Print pattern

(i) #include<iostream>

using namespace std;

```
int main() {  
    int i,j,n;  
    cout<<"enter the number";  
    cin>>n;  
    for(i=5;i>=1;i--){  
        for(j=1;j<=i;j++){  
            cout<<"*";  
        }  
        cout<<endl;  
    }  
    return 0;  
}
```

**

*

(ii) #include<iostream>

using namespace std;

int main() {

int rows;

cout<<"enter rows:";

cin>>rows;

for (int i = rows; i >= 1; i--) {

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

if(j<i){

cout << " ";

}else{

cout<<"* ";

}

}

cout << endl;

}

return 0;

}

*

* *

* * *

* * * *

* * * * *

(iii) #include<iostream>

using namespace std;

```
int main() {  
    int n;  
    cout<<"enter the number";  
    cin>>n;  
    for(int i=1;i<=n;i++){  
        for(int j=1;j<=i;j++){  
            cout<<i;  
        }  
        cout<<endl;  
    }  
    return 0;  
}
```

1

22

333

4444

55555

```
(v) #include<iostream>
```

```
using namespace std;
```

```
int main(){
```

```
    int n;
```

```
    cout<<"enter the number:";
```

```
    cin>>n;
```

```
    int coun=1;
```

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

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

```
            cout<<coun<<" ";
```

```
            coun++;
```

```
        }
```

```
        cout<<endl;
```

```
    }
```

```
    return 0;
```

```
}
```

```
1
```

```
2 3
```

```
4 5 6
```

```
7 8 9 10
```

```
11 12 13 14 15
```

Q16. (i)

```
#include<iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    cout<<"the numbers which are palindrome are:"<<endl;
```

```
    for(int num=500;num<=1000;num++){
```

```
        int rev=0;
```

```
        int original;
```

```
        original=num;
```

```
        int temp = num;
```

```
        while(temp>0){
```

```
            int r = temp%10;
```

```
            temp = temp/10;
```

```
            rev = (rev*10) + r;
```

```
        }
```

```
        if(original==rev){
```

```
            cout<<original<<endl;
```

```
        }
```

```
    }
```

```
    return 0;
```

```
}
```

(ii) to print first 50 odd numbers

```
#include<iostream>

#include<climits>

using namespace std;

int main() {

    int count=0;

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

        if(i%2!=0){

            cout<<i<<" ";

            count++;

        }

        if(count==50)

            break;

    }

    return 0;

}
```

(iii) to find binary , decimal, hexadecimal equivalent of a decimal.

```
#include <iostream>

using namespace std;

void binary(int n) {
    int original = n;
    int pow = 1;
    int ans = 0;
    while (n != 0)
    {
        int rem = n % 2;
        ans = ans + rem * pow;
        n = n / 2;
        pow = pow * 10;
    }
    cout << "The binary number for " << original << " is " << ans << "\n";
}

void octal(int n)
{
    int original = n;
    int pow = 1;
    int ans = 0;
    while (n != 0)
    {
        int rem = n % 8;
        ans = ans + rem * pow;
        n = n / 8;
        pow = pow * 10;
    }
    cout << "The octal number for " << original << " is " << ans << "\n";
}

void hexadecimal(int n)
{

```

```
int original = n;

int pow = 1;

int ans = 0;

while (n != 0)
{
    int rem = n%16;

    ans = ans + rem * pow;

    n = n / 16;

    pow = pow * 10;
}

cout << "The hexadecimal number for " << original << " is " << ans << "\n";
}

int main()
{
    int num;

    cout << "Enter a number:";

    cin >> num;

    binary(num);

    octal(num);

    hexadecimal(num);

    return 0;
}
```

(iv) to find decimal equivalent of given binary hexadecimal,octal.

```
#include<iostream>
```

```
#include<cmath>
```

```
using namespace std;
```

```
void bin(int n){
```

```
    int power=0,ans=0,rem;
```

```
    while (n > 0)
```

```
    { int rem = n%10;
```

```
        ans = ans + rem * pow(2, power);
```

```
        n = n / 10;
```

```
        power++;
```

```
    }
```

```
    cout<<"the decimal number is:"<<ans<<endl;
```

```
}
```

```
void oct(int n){
```

```
    int rem,power=0,oct=0;
```

```
    while(n>0){
```

```
        int rem = n%10;
```

```
        oct = oct + rem*pow(8,power);
```

```
        n = n/10;
```

```
        power++;
```

```
    }
```

```
    cout<<"the decimal is:"<<oct;
```

```
}
```

```
void hexa(int n){
```

```
    int rem ,power=0,hexa=0;
```

```
    while(n>0){
```

```
        int rem = n%10;
```

```

        hexa = hexa + rem*pow(16,power);

        n= n/10;

        power++;

    }

    cout<<"the decimal is:"<<hexa;

}

int main()
{
    int num, choice = 0;

    cout << " 1 if number is binary\n 2 if it is octal \n 3 if it is hexadecimal:"<<endl;
    cin >> choice;

    cout << "Enter a number:";

    cin >> num;

    if (choice == 1)
    {
        bin(num);
    }

    else if (choice == 2)
    {
        oct(num);
    }

    else if (choice == 3)
    {
        hexa(num);
    }

    else
    {
        cout << "Invalid choice.";
    }

    return 0;

}

```


(v) to find geometrical sum upto n terms

```
#include<iostream>

#include<cmath>

using namespace std;

int main() {

    float sum=0,a,r,n;

    cout<<"enter the first term:";

    cin>>a;

    cout<<"enter ratio:";

    cin>>r;

    cout<<"enter number of terms:";

    cin>>n;

    sum = a*(pow(r,n)-n)/r-1;

    cout<<"the geometric sum is:"<<sum;

    return 0;

}
```

Q17 (i) print binary from decimal

```
#include<iostream>
```

```
using namespace std;
```

```
int decimalToBinary(int n){
```

```
    int ans=0;
```

```
    int pow=1;
```

```
    while(n>0){
```

```
        int rem= n%2;
```

```
        n=n/2;
```

```
        ans+=(rem*pow);
```

```
        pow = pow*10;
```

```
    }
```

```
    return ans;
```

```
}
```

```
int main(){
```

```
    int n;
```

```
    cout<<"enter the number:";
```

```
    cin>>n;
```

```
    cout<<decimalToBinary(n);
```

```
    return 0;
```

```
}
```

(ii) to print octal number of a given decimal number.

```
#include<iostream>
```

```
using namespace std;
```

```
int decimalToOctal(int n){
```

```
    int rem, ans=0;
```

```
    int i=1;
```

```
    while(n>0){
```

```
        rem=n%8;
```

```
        n=n/8;
```

```
        ans= ans + (rem*i);
```

```
        i=i*10;
```

```
    }
```

```
    return ans;
```

```
}
```

```
int main(){
```

```
    int n;
```

```
    cout<<"enter the number:";
```

```
    cin>>n;
```

```
    cout<<decimalToOctal(n);
```

```
    return 0;
```

```
}
```

(iii) to find factorial

```
#include<iostream>
```

```
using namespace std;
```

```
int fact(int n){
```

```
    int ans=1;
```

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

```
        ans = ans*i;
```

```
    }
```

```
    return ans;
```

```
}
```

```
int main(){
```

```
    int n;
```

```
    cout<<"enter the number:";
```

```
    cin>>n;
```

```
    cout<<fact(n);
```

```
    return 0;
```

```
}
```

(iv) to n terms of Fibonacci series

```
#include<iostream>

using namespace std;
```

```
int feb(int n){
    cout<<"0 1 ";

    int a=0 ,b=1,c;
    for(int i=1;i<=n;i++){
        c=a+b;
        a=b;
        b=c;
        cout<<c<<" ";
    }
    return 0;
}
```

```
int main(){
    int n;
    cout<<"enter the number:";
    cin>>n;
    cout<<feb(n);
    return 0;
}
```

Q18. Wap to find the avg of 1D array.

```
#include <iostream>

using namespace std;

int main(){
    int n,sum=0;
    cout<<"enter number of elements";
    cin>>n;
    float arr[n];
    cout<<"enter elements";
    for(int i=1;i<=n;i++){
        cin>>arr[i];
        sum = sum + arr[i];
    }
    float avg = sum/n;
    cout<<"the average is:"<<avg;
    return 0;
}
```

Q19. To find max and min element from a 1D array.

```
#include <iostream>
```

```
#include<climits>
```

```
using namespace std;
```

```
int main(){
```

```
    int arr[5];
```

```
    int min=INT_MAX,max=INT_MIN;
```

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

```
        cout<<"enter elements:";
```

```
        cin>>arr[i];
```

```
        if(arr[i]<min){
```

```
            min=arr[i];
```

```
        }
```

```
        if(arr[i]>max){
```

```
            max=arr[i];
```

```
        }
```

```
    }
```

```
    cout<<max<<endl;
```

```
    cout<<min;
```

```
    return 0;
```

```
}
```

Q20. To print transpose of a matrix

```
#include <iostream>
```

```
using namespace std;
```

```
int main(){
```

```
    int rows=3,column=2;
```

```
    int M[rows][column]={1,2},{3,4},{5,6}};
```

```
    int T[column][rows];
```

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

```
        for(int j=0;j<column;j++){
```

```
            T[j][i]=M[i][j];
```

```
        }
```

```
    }
```

```
    cout<<"the matrix is:"<<endl;
```

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

```
        for(int j=0;j<column;j++){
```

```
            cout<<M[i][j]<<" ";
```

```
        }
```

```
        cout<<endl;
```

```
    }
```

```
    cout<<"the transpose is:"<<endl;
```

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

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

```
            cout<<T[i][j]<<" ";
```

```
        }
```

```
        cout<<endl;
```

```
    }
```

```
    return 0;
```

```
}
```


Q21 WAP to add to two matrix

```
#include<iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int rows, columns;
```

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

```
    cin >> rows;
```

```
    cout << "Enter the number of columns: ";
```

```
    cin >> columns;
```

```
    int A[rows][columns], B[rows][columns], C[rows][columns];
```

```
    cout << "Enter elements of Matrix A:" << endl;
```

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

```
        for(int j = 0; j < columns; j++) {
```

```
            cout<<i<<" "<<j;
```

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

```
        }
```

```
    }
```

```
    cout << "Enter elements of Matrix B:" << endl;
```

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

```
        for(int j = 0; j < columns; j++) {
```

```
            cout<<i<<" "<<j;
```

```
            cin >> B[i][j];
```

```
        }
```

```
    }
```

```
for(int i = 0; i < rows; i++) {  
    for(int j = 0; j < columns; j++) {  
        C[i][j] = A[i][j] + B[i][j];  
    }  
}  
  
cout << "Resultant Matrix after Addition:" << endl;  
for(int i = 0; i < rows; i++) {  
    for(int j = 0; j < columns; j++) {  
        cout << C[i][j] << " ";  
    }  
    cout << endl;  
}  
  
return 0;  
}
```

Q22 wap for multiplication of two matrix

```
#include<iostream>
```

```
using namespace std;
```

```
int main(){
```

```
int row, column;
```

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

```
cin >> row;
```

```
cout << "Enter the number of columns: ";
```

```
cin >> column;
```

```
int A[row][column],B[row][column],C[row][column];
```

```
cout<<"enter matrix A"<<endl;
```

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

```
    for(int j=0;j<column;j++){
```

```
        cout<<i<<" "<<j;
```

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

```
    }
```

```
}
```

```
cout<<"enter matrix B"<<endl;
```

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

```
    for(int j=0;j<row;j++){
```

```
        cout<<i<<" "<<j;
```

```
        cin>>B[i][j];
```

```
    }
```

```
}
```

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

```
    for(int j=0;j<column;j++){
```

```
        C[i][j]=A[i][j]*B[j][i];
```

```
    }  
}  
  
cout<<"Resultant matrix is:"<<endl;  
for(int i=0;i<row;i++){  
    for(int j=0;j<column;j++){  
        cout<<C[i][j]<<" ";  
    }  
    cout<<endl;  
}  
return 0;  
}
```

Tejasv Patel (IT-2k24-89)

Q23 wap to sort an array in ascending order

```
#include <iostream>
```

```
using namespace std;
```

```
void sort(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]){
```

```
                swap(arr[j],arr[j+1]);
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

```
void printarray(int arr[],int n){
```

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

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

```
    }
```

```
    cout<<endl;
```

```
}
```

```
int main(){
```

```
    int arr[]={2,6,2,7,3,9};
```

```
    int n = 6;
```

```
    cout<<"original array:";
```

```
    printarray(arr,n);
```

```
    sort(arr,n);
```

```
    cout<<"sorted matrix:";
```

```
printarray(arr,n);
```

```
return 0;
```

```
}
```

Tejasv Patel (IT-2k24-89)

Q24 wap to reverse a given string

```
#include<iostream>
```

```
using namespace std;
```

```
int main(){
```

```
    string rev,str;
```

```
    cout<<"enter the string:";
```

```
    cin>>str;
```

```
    for(int i=str.length()-1;i>=0;i--)
```

```
    {
```

```
        rev = rev + str[i];
```

```
    }
```

```
    cout<<"the reverse is:"<<rev;
```

```
    return 0;
```

```
}
```

Q25.WAP to count all the vowels in the string.

```
#include <iostream>

using namespace std;

int main()
{
    int count=0;

    string str;

    cout<<"enter the string:";

    cin>>str;

    for(int i=0;i<str.length();i++){
        char ch=str[i];

        if(ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u' || ch=='A' || ch=='E' || ch=='I' || ch=='O' || ch=='U'){
            count++;
        }
    }

    cout<<"the number of vowels are:"<<count;

    return 0;
}
```


Q26.WAP to check the given string is palindrome or not?

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    string rev,ori,str;
```

```
    cout<<"enter the string:";
```

```
    cin>>str;
```

```
    ori=str;
```

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

```
        rev = rev + str[i];
```

```
    }
```

```
    if(rev==ori){
```

```
        cout<<"the string is palindrome";
```

```
    }
```

```
    else{
```

```
        cout<<"not palindrome";
```

```
    }
```

```
    return 0;
```

```
}
```

Q28. 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
{
private:

    string make,model;

    int year;
public:

    void inputdata(){

        cout<<"enter car detials:"<<endl;
        cout<<"make:";
        cin>>make;
        cout<<"model:";
        cin>>model;
        cout<<"year of manufacture:";
        cin>>year;
    }

    void output(){

        cout<<"car detials:"<<endl;
        cout<<"make:"<<make<<endl;
        cout<<"model:"<<model<<endl;
        cout<<"year:"<<year<<endl;
    }

};
```

```
int main(){  
    car mycar;  
    mycar.inputdata();  
    mycar.output();  
  
    return 0;  
}
```

Tejasv Patel (IT-2k24-89)

Q29. 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 person;
```

```
class address{
    string street;
    string city;
    int zipcode;
```

```
public:
```

```
void setdata(string s,string c,int z){
    street=s;
    city=c;
    zipcode=z;
}
```

```
string getstreet(){
    return street;
}
```

```
string getcity(){
    return city;
}
```

```
int getzipcode(){
    return zipcode;
```

```
}
```

```
};
```

```
class person{
```

```
    address a;
```

```
    public:
```

```
    void takedata(address a1){
```

```
        a = a1;
```

```
    }
```

```
    void getstreet(){
```

```
        cout<<"street:"<<a.getstreet();
```

```
    }
```

```
    void getcity(){
```

```
        cout<<"city:"<<a.getcity();
```

```
    }
```

```
    void getzipcode(){
```

```
        cout<<"zipcode:"<<a.getzipcode();
```

```
    }
```

```
};
```

```
int main(){
```

```
    address a1;
```

```
    a1.setdata("main road","indore",404040);
```

```
    person p1;
```

```
    p1.takedata(a1);
```

```
    p1.getcity();
```

```
        cout<<endl;
```

```
    p1.getstreet();
```

```
        cout<<endl;
```

```
p1.getzipcode();
```

```
cout<<endl;
```

```
return 0;
```

```
}
```

Tejasv Patel (IT-2k24-89)

Q30. Write a program to display the minimum, maximum, sum, search and average of elements of an array.

```
#include <iostream>

using namespace std;

int main() {

    int n;

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

    cin >> n;

    int arr[n];

    cout << "Enter the elements: ";

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

        cin >> arr[i];

    }

    int minVal = arr[0];

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

        if (arr[i] < minVal) {

            minVal = arr[i];

        }

    }

    int maxVal = arr[0];

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

        if (arr[i] > maxVal) {

            maxVal = arr[i];

        }

    }

    int sum = 0;

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

```
        sum += arr[i];
    }

    int search;
    cout << "Enter the value to search: ";
    cin >> search;
    int x=0;
    for (int i = 0; i < n; i++) {
        if (arr[i] == search) {
            x=1;
            break;
        }
    }
    if (x==1) {
        cout << "Value found in the array." << endl;
    } else {
        cout << "Value not found in the array." << endl;
    }

    double average = (double)sum / n;

    cout << "Minimum value: " << minVal << endl;
    cout << "Maximum value: " << maxVal << endl;
    cout << "Sum: " << sum << endl;
    cout << "Average: " << average << endl;
    return 0;
}
```


Q31. 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>
```

```
using namespace std;
```

```
class student{
```

```
    int admno;
```

```
    char sname[20];
```

```
    float eng, maths, science;
```

```
    float total;
```

```
public:
```

```
    float ctotal(){
```

```
        total = eng + maths + science;
```

```
        return total;
```

```
    }
```

```
void takedata(){
```

```
    cout<<"enter adimission number: ";
```

```
    cin>> admno;
```

```
    cout<< "student name:";
```

```
    cin>> sname;
```

```
    cout<< "english marks:";
    cin>> eng;
    cout<<"maths marks";
    cin>> maths;
    cout<< " science marks:";
    cin>>science;
}
```

```
void showdata(){
    cout<< "adimission number:"<<admno<<endl;
    cout<<"student name:"<<sname<<endl;
    cout<<"english marks:"<<eng<<endl;
    cout<<"maths marks:"<<maths<<endl;
    cout<<"science marks:"<<science<<endl;
    cout<<"total:"<<ctotal();
}
};
```

```
int main(){
    student s1;
    s1.takedata();
    s1.showdata();
    return 0;
}
```

Q32. 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

<u>Distance</u>	<u>Fuel</u>
<u><=1000</u>	<u>500</u>
<u>more than 1000 and <=2000</u>	<u>1100</u>
<u>more than 2000</u>	<u>2200</u>

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{
```

```
    int flightno;
```

```
    string destination;
```

```
    float distance;
```

```
    int fuel;
```

```
    void calfuel(){
```

```
        int distance;
```

```
        if(distance<=1000){
```

```
            fuel = 1000;
```

```
        }
```

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

```

        fuel= 1100;
    }
    else{
        fuel= 2000;
    }
}

public:
    void feedinfo(){
        cout<<"enter flightno.: ";
        cin>>flightno;
        cout<<"enter destination: ";
        cin>>destination;
        cout<<"enter distance: ";
        cin>>distance;
        calfuel();
    }

    void showinfo(){
        cout<<"enter flightno.: "<<flightno<<endl;
        cout<<"enter destination: "<<destination<<endl;
        cout<<"enter distance: "<<distance<<endl;
        cout<<"fuel: "<<fuel<<endl;
    }
};

int main(){
    flight flight;
    flight.feedinfo();
    flight.showinfo();
    return 0;
}

```

Q33. Write a menu driven program to perform following:

- a) **Input a matrix**
- b) **Display matrix**
- c) **Add two matrix**
- d) **Multiply two matrix**
- 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 elements of the matrix " << rows << "x" << cols ;
    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 c2) {
```

```

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 res[MAX][MAX], int rows, int cols) {
    for (int i = 0; i < rows; i++)
        for (int j = 0; j < cols; j++)
            res[j][i] = mat[i][j];
}

int main() {
    int mat1[MAX][MAX], mat2[MAX][MAX], res[MAX][MAX];
    int r1, c1, r2, c2;
    int choice;
    do {
        cout << "\nMenu:\n";
        cout << "1. Input a matrix\n";
        cout << "2. Display matrix\n";
        cout << "3. Add two matrices\n";
        cout << "4. Multiply two matrices\n";
        cout << "5. Transpose a matrix\n";
        cout << "6. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;
        switch (choice) {
            case 1:
                cout << "Enter rows and columns of matrix: ";
                cin >> r1 >> c1;
                inputMatrix(mat1, r1, c1);

```

```
break;
```

case 2:

```
displayMatrix(mat1, r1, c1);
```

```
break;
```

case 3:

```
cout << "Enter rows and columns of both matrices: ";
```

```
cin >> r1 >> c1;
```

```
inputMatrix(mat1, r1, c1);
```

```
inputMatrix(mat2, r1, c1);
```

```
addMatrix(mat1, mat2, res, r1, c1);
```

```
displayMatrix(res, r1, c1);
```

```
break;
```

case 4:

```
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!\n";
```

```
    break;
```

```
}
```

```
inputMatrix(mat1, r1, c1);
```

```
inputMatrix(mat2, r2, c2);
```

```
multiplyMatrix(mat1, mat2, res, r1, c1, c2);
```

```
displayMatrix(res, r1, c2);
```

```
break;
```

case 5:

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

```
cin >> r1 >> c1;
```

```
inputMatrix(mat1, r1, c1);
```

```
transposeMatrix(mat1, res, r1, c1);
```

```
        displayMatrix(res, c1, r1);

        break;

    case 6:

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

        break;

    default:

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

    }

} while (choice != 6);

return 0;

}
```