--- C++ ASSIGNMENT ---

Q1.) Write a program to find whether the given number is even or odd.

SOL-:

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

int main (){
    int n;
    cout << "enter the no to check it is even or odd: ";
    cin >> n;

    if(n % 2 == 0){
        cout << "the given no. is even";
    }
    else{
        cout << "the given no. is odd";
    }
}
```

Q2.) Write a program to find whether the given number is prime or composite.

SOL-:

```
//to find out if the given number is prime or composite.
#include <iostream>
using namespace std;
int main ( )
{
int n;
cout<< “ENTER THE NUMBER”: ;
cin>>n;
for (int i = 2; i <= n; i++)
{
if(n % i != 0)
continue;
else if ( n % i == 0 && n == i)
cout<< “THE NUMBER IS PRIME”;
```

```
else
cout<< “THE NUMBER IS COMPOSITE”;
break;
}
return 0;
}
```

Q3.) Write a program to print table of a number upto ‘N’ multiples.

SOL-:

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

int main(){
    int n , i;
    cout << "enter the no. for which table has to be displayed: ";
    cin >> n;
```

```
    cout << "enter to which multiply: ";
    cin >> i;

    for(int j = 1 ; j <= i ; j++){
        cout << n << " x " << j <<  " = " << j*n << endl;
    }
}
```

Q4.) WAP to find

i) greatest of two no.'s

ii) greatest of three no.'s

SOL-:

```
// for two numbers-
#include <iostream>
using namespace std;
int main(){
int a,b;
cout<<"enter two numbers";
cin>>a>>b;
if(a>b){
```

```
cout<<" is greater;
}
else{
cout<<b<<" is greater;}
return 0;
}
// for three numbers-
#include <iostream>
using namespace std;
int main(){
int a,b,c;
cout<<"enter three numbers";
cin>>a>>b>>c;
if(a>b){
  if(a>c){
cout<<"a is greatest";
}
else if (a<c){
cout<<" c is greatest;
```

```
}else if (b>c){
cout<<”b is greatest;
}
else{
cout<<” is greatest;
}
return 0;
}
```

Q5.)WAP to find sum of first 'n' natural no.'s

SOL-:

```
 #include <iostream>
using namespace std;
int main (){
    int n , sum = 0;
    cout << "enter the nth number to find sum: ";
    cin >> n;

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

```
    }
    cout << "the sum of first nth natural no. " << sum;
}
```

Q6.)WAP to find factorial of given no.

SOL:-

```
// to find the factorial of a given number.
#include<iostream>
using namespace std;
int main ( )
{
int n, i, a = 1;
cout<< “ENTER THE NUMBER WHOSE FACTORIAL IS TO BE CALCULATED: ”;
cin>> n;
for (i = 1; i <= n; i++)
{
a = a * i;
}
```

```
cout<< “THE FACTORIAL COMES OUT TO BE: ”<< a;
return 0;
}
```

Q7.)WAP to find sum of digits of 'n' digit no.

SOL-:

```
#include <iostream>
using namespace std;
int main(){
    int n , sum =0;
    cout << "enter the no to find the sum of all digit: ";
    cin >> n;
 while(n > 0){
        int r = n%10;
 sum = sum + r;
        n = n/10;
    }
    cout << "the sum of all the digits is " << sum;
}
```

Q8.)WAP to find reverse of a no.

SOL:-

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

int main(){
    int n , rev = 0;
    cout << "enter the no. to reverse: ";
    cin >> n;

    while(n > 0){
        int r = n%10;
        rev = (rev*10) + r;
        n = n/10;
    }

    cout << "the reverse of the no is " << rev;
}
```

Q9.)WAP to determine given no. is palindrome or not.

SOL-:

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

int main(){
    int n , rev = 0 , original;
    cout << "enter the no. to check: ";
    cin >> n;

    original = n;

    while(n > 0){
        int r = n % 10;
        rev = (rev * 10) + r;
        n = n /10;
}
 if(original == rev){
        cout << "the given no. is palindrome";
    }
    else{
```

```
        cout << "the given no. is not a palindrome";
    }
}
```

Q10.)WAP to print Fibonacci series up to 'n' terms.

SOL-:

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

int main(){
    int n, a = 0 , b = 1 , temp;
    cout << "enter upto which term: ";
    cin >> n;

    for(int i = 0 ; i <= n ; i++){
        temp = a + b;
        cout << temp << endl;
        a = b;
```

```
        b = temp;
    }
}
```

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

SOL-:

```
include <iostream>
using namespace std;

int power(int base , int power){
    int pow = 1;
    for(int i = 0 ; i < power ; i++){
        pow = pow * base;
    }
    return pow;
}

int digit(int n){
    int dig = 0;
```

```
    while(n > 0){
        dig++;
        n /= 10;
    }
    return dig;;
}

int main(){
    int n , arm = 0;
    int org;
    cout << "enter the no to check its armstrong or not: ";
    cin >> n;

    int pow = digit(n);
    org = n;

    while(n > 0){
        int r = n%10;
        arm = power(r,pow) + arm;
```

```
        n /=10;
    }

    if(org == arm){
        cout << "the given no. is armstrong";
    }
    else{
        cout << "the given no. is not armstrong ";
    }
}
```

Q12.)WAP to print all even no.'s between 100 & 200.

SOL:-

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

int main(){
    for(int i = 101 ; i <200 ; i++){
        if(i % 2 == 0){
            cout << i << endl;
```

```
        }
        else{
            continue;
        }
    }
}
```

Q13.)WAP to print first 50 prime numbers.

SOL:-

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

int main ()
{
    int count = 1;

    for(int i = 2 ; i < 1000 ; i++)
    {
        int m = 0;
        for(int j = 2 ; j < i ; j++)
```

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

               m++;
            }
        }
        if(m == 0)
        {
            cout << count << ". " << i << endl;
            count++;
        }
        if( count >= 51){
            break;
        }
    }
}
```

Q14.)WAP to print all 4 digit Armstrong no.'s.

SOL-:

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

int power(int base , int power){
    int pow = 1;
    for(int i = 0 ; i < power ; i++){
        pow = pow * base;
    }
    return pow;
}

int main(){

    for(int i = 1000 ; i <=9999 ; i++){
        int org;
        int n = i;
        org = n;
        int arm = 0;
```

```
        while(n > 0){
            int r = n% 10;
            arm = power(r,4) + arm;
            n /= 10;
        }
        if(arm == org){
            cout << arm << endl;
        }
        else{
            continue;
        }
    }
}
```

Q15.)WAP to print following patterns:

i)

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

.....

.....

SOL-:

```
 #include <iostream>
  using namespace std;
  void main(){
   for(int i = 0 ; i < 5 ; i++){
       for(int j = 0 ; j <= i ; j++){
           cout << "*";
       }
       cout << "\n";
   }
}
```

.....

ii)

*****

****

***

**

*

SOL-:

```
#include <iostream>
  using namespace std;
  void main(){
   for(int i = 5 ; i > 0 ; i--){
       for(int j = 0 ; j < i ; j++){
           cout << "*";
       }
       cout << "\n";
   }
}
```

iii)

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

SOL:-

```
#include <iostream>
  using namespace std;
  void main(){
   for(int i = 1 ; i <= 3 ; i++){
       for(int j = 5-i ; j > 0 ; j--){
           cout << " ";
       }
       for(int k = 1 ; k <= (2*i)- 1 ; k++){
           cout << "*";
       }
       cout << "\n";
   }

}
```

iv)

1

22

333

4444

SOL:-

```
#include <iostream>
   using namespace std;
   void main(){
    for(int i = 1 ; i <= 5 ; i++){
        for(int j = 1 ; j <= i ; j++){
            cout << i;
        }
        cout << "\n";
    }
}
```

v) Pascal's triangle

SOL:-

```
 #include<iostream>
using namespace std;
int fact(int x){
```

```
    int f=1;
    for(int i=1;i<=x;i++){
        f*=i;
    }
    return f;
}
int combi(int n,int r){
    int c;
    c= fact(n) /(fact(r) * fact(n-r) );
    return c;
}
int main() {
    int n;
    cout<<"Enter a number: ";
    cin>>n;
    for(int i=0;i<=n;i++){
        for(int k=0;k<n-i;k++){
            cout<<" ";
        }
```

```
        for(int j=0;j<=i;j++){
            cout<<combi(i,j)<<" ";
        }
        cout<<endl;
    }
    return 0;
}
```

vi) Floyd's triangle

SOL-:

```
    #include <iostream>
   using namespace std;
   void main(){
    int count = 1;
    for(int i = 1 ; i < 5 ; i++ ){
        for(int j = 1 ; j <= i ; j++){
            cout << count;
            count++;
        }
        cout << "\n";
```

```
    }
}
```

16.) Using functions, write the following C++ programs.

i) To print all palindromes for a range 500-1000.

SOL:-

```
#include <iostream>
using namespace std;
int main()
{
 int num, orig, rev = 0;
    cout << "Palindrome numbers between 500 and 1000 are:" << endl;
    for (int num = 500; num <= 1000; num++)
    {
      orig = num;
       rev = 0;
       while (orig != 0)
       {
```

```
            int digit = orig % 10;
            rev = rev * 10 + digit;
            orig /= 10;
        }
        if (rev == num)
        {
            cout << num<<",";
        }
    }
    return 0;
}
```

ii) To print first 100 odd numbers.

SOL-:

```
#include <iostream>
using namespace std;
void Odd()
{
   int n;
  for (n=1; n<=200;n++)
```

```
  {
    if (n%2!=0)
      {
         cout << "\n"<<n;
      }
  }
}
int main()
{
  Odd();
  return0;
}
```

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

SOL-:

```
#include <iostream>
using namespace std;
void binary(int n)
```

```
{
    int org_num = n;
    int factor = 1;
    int bin = 0;
    while (n != 0)
    {
        bin = bin + (n % 2) * factor;
        n = n / 2;
        factor = factor * 10;
    }
    cout << "The binary number for " << org_num << " is "
<< bin << "\n";
}
void octal(int n)
{
    int org_num = n;
    int factor = 1;
    int oct = 0;
    while (n != 0)
```

```
    {
        oct = oct + (n % 8) * factor;
        n = n / 8;
        factor = factor * 10;
    }
    cout << "The octal number for " << org_num << " is " <<
oct << "\n";
}
void hexadecimal(int n)
{
    int org_num = n;
    int factor = 1;
    int hexa = 0;
    while (n != 0)
    {
        hexa = hexa + (n % 16) * factor;
        n = n / 16;
        factor = factor * 10;
    }
```

```
    cout << "The hexadecimal number for " << org_num <<
" is " << hexa << "\n";
}
int main()
{
    int num;
    cout << "Enter a number:";
    cin >> num;
    binary(num);
    octal(num);
    hexadecimal(num);
    return 0;
}
```

iv) To find decimal equivalents for given binary, hexadecimal and octal numbers.

SOL-:

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

```
void bin(int n)
{
    int org_num = n;
    int deci = 0;
    int power = 0;
    while (n != 0)
    {
        deci = deci + (n % 10) * pow(2, power);
        n = n / 10;
        power++;
    }
    cout << "The decimal number for binary " << org_num
<< " is " << deci << "\n";
}
void oct(int n)
{
    int org_num = n;
    int deci = 0;
    int power = 0;
```

```
    while (n != 0)
    {
        deci = deci + (n % 10) * pow(8, power);
        n = n / 10;
        power++;
    }
    cout << "The decimal number for octal " << org_num <<
" is " << deci << "\n";
}
void hex(int n)
{
    int org_num = n;
    int deci = 0;
    int power = 0;
    while (n != 0)
    {
        deci = deci + (n % 10) * pow(16, power);
        n = n / 10;
        power++;
```

```
    }
    cout << "The decimal number for hexadecimal " << org_num << " is " << deci << "\n";
}
int main()
{
    int num, choice = 0;
    cout << "Enter 1 if number is binary, 2 if it is octal and 3 if it is hexadecimal:";
    cin >> choice;
    cout << "Enter a number:";
    cin >> num;
    if (choice == 1)
    {
        bin(num);
    }
    else if (choice == 2)
    {
        oct(num);
```

```
    }
    else if (choice == 3)
    {
        hex(num);
    }
    else
    {
        cout << "Invalid choice.";
    }
    return 0;
}
```

v) To calculate geometric sum upto n terms.

SOL-:

```
#include <iostream>
using namespace std;
double geometricSum(double a, double r, int n) {
    double sum = 0;
    double term = a;
    for (int i = 0; i < n; i++) {
```

```
        sum = sum+term;
        term = term* r;
    }
    return sum;
}
int main() {
    double a, r;
    int n;
    cout << "Enter the first term (a): ";
    cin >> a;
    cout << "Enter the common ratio (r): ";
    cin >> r;
    cout << "Enter the number of terms (n): ";
    cin >> n;
    double sum = geometricSum(a, r, n);
    cout << "Geometric sum up to " << n << " terms: " <<
sum << endl;
    return 0;
}
```

Q17. Using recursion write following c++ programs.

i) Print binary number for a decimal number

```
#include <iostream>
using namespace std;
void toBin(int n)
{
    if (n > 1)
    {
        toBin(n / 2);
    }
    cout << n % 2;
}
int main()
{
    int num;
    cout << "Enter the number:";
    cin >> num;
    if (num == 0)
    {
```

```
        cout << 0;
    }
    else
        toBin(num);
return0;
}
```

ii) Print octal number for a decimal number.

```
#include <iostream>
using namespace std;
void toOct(int n)
{
    if (n > 1)
    {
        toOct(n / 8);
    }
    cout << n % 8;
```

```
}
int main()
{
    int num;
    cout << "Enter the number:";
    cin >> num;
    if (num == 0)
    {
        cout << 0;
    }
    else
        toOct(num);
    return 0;
}
```

iii) Print factorials for a given range.

```
#include <iostream>
using namespace std;
int factorial(int n)
{
```

```
    if (n == 0 || n == 1)
    {
        return 1;
    }
    else {
        return n * factorial(n - 1);
    }
}
int main() {
    int upper_limit, lower_limit;
    cout << "For range of factorials, enter lower limit:";
    cin >> lower_limit;
    cout << "Enter upper limit:";
    cin >> upper_limit;
    for (int i = lower_limit; i <= upper_limit; i++)  {
        int fact;
        fact = factorial(i);
        cout << "The factorial of " << i << " is " << fact << "\n";
    }
```

```
    return 0;
}
```

iv) Print first n terms of fibonacci series.

```
#include <iostream>
using namespace std;
int fibonacci(int n)
{
    if (n == 0)
    {
        return 0;
    }
    else if (n == 1)  {
        return 1;
    }
    else   {
        return fibonacci(n - 1) + fibonacci(n - 2);
    }
}
int main()  {
```

```
    int n;
    cout << "Enter the number of terms for fibonacci
series:";
    cin >> n;
    cout << "Fibonacci series:\n";
    for (int i = 0; i < n; i++) {
        cout << fibonacci(i) << " ";
    }
    return 0;
}
```

Q18. WAP to find average of all the elements of 1D 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 sum = 0;
    for (int i = 0; i < n; i++)
   {
        sum += arr[i];
    }
    double average = (double)sum / n;
    cout << "Average: " << average << endl;
    return 0;
}
```


Q19. WAP to find maximum and minimum value of a 1D numeric 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 maxVal = arr[0];
    int minVal = arr[0];
    for (int i = 1; i < n; i++) {
        if (arr[i] > maxVal) {
            maxVal = arr[i];
        }
        if (arr[i] < minVal) {
            minVal = arr[i];
```

```
        }
    }
    cout << "Maximum value: " << maxVal << endl;
    cout << "Minimum value: " << minVal << endl;
    return 0;
}
```

Q20. Write a program to find transpose of a 2D matrix.

```
#include <iostream>
using namespace std;
int main()
{
    int rows, cols;
    cout << "Enter number of rows: ";
    cin >> rows;
    cout << "Enter number of columns: ";
    cin >> cols;
    int matrix[rows][cols];
    cout << "Enter matrix elements: " << endl;
```

```
for (int i = 0; i < rows; i++)
 {
    for (int j = 0; j < cols; j++)
     {
        cin >> matrix[i][j];
     }
  }
cout << "Original Matrix: " << endl;
for (int i = 0; i < rows; i++)
 {
    for (int j = 0; j < cols; j++)
    {
        cout << matrix[i][j] << " ";
    }
    cout << endl;
}
cout << "Transpose Matrix: " << endl;
for (int i = 0; i < cols; i++)
  {
```

```
        for (int j = 0; j < rows; j++)
        {
            cout << matrix[j][i] << " ";
        }
        cout << endl;
    }
    return 0;
}
```

Q21. WAP to add two matrices.

```
#include <iostream>
using namespace std;
int main() {
    int rows, cols;
    cout << "Enter number of rows: ";
```

```
cin >> rows;
cout << "Enter number of columns: ";
cin >> cols;
int matrix1[rows][cols];
int matrix2[rows][cols];
cout << "Enter elements of Matrix 1: " << endl;
for (int i = 0; i < rows; i++) {
    for (int j = 0; j < cols; j++) {
        cin >> matrix1[i][j];
    }
}
cout << "Enter elements of Matrix 2: " << endl;
for (int i = 0; i < rows; i++) {
    for (int j = 0; j < cols; j++) {
        cin >> matrix2[i][j];
    }
}
cout << "Matrix 1: " << endl;
for (int i = 0; i < rows; i++) {
```

```
    for (int j = 0; j < cols; j++) {
        cout << matrix1[i][j] << " ";
    }
    cout << endl;
}
cout << "Matrix 2: " << endl;
for (int i = 0; i < rows; i++) {
    for (int j = 0; j < cols; j++) {
        cout << matrix2[i][j] << " ";
    }
    cout << endl;
}
cout << "Sum of Matrix 1 and Matrix 2: " << endl;
for (int i = 0; i < rows; i++) {
    for (int j = 0; j < cols; j++) {
        cout << matrix1[i][j] + matrix2[i][j] << " ";
    }
    cout << endl;
}
```

```
    return 0;
}
```

Q22. WAP to multiply 2D matrices.

```
#include <iostream>
using namespace std;
int main() {
    int rows1, cols1, rows2, cols2;
    cout << "Enter number of rows for Matrix 1: ";
    cin >> rows1;
    cout << "Enter number of columns for Matrix 1: ";
    cin >> cols1;
    cout << "Enter number of rows for Matrix 2: ";
    cin >> rows2;
    cout << "Enter number of columns for Matrix 2: ";
```

```
    cin >> cols2;
    if (cols1 != rows2) {
        cout << "Matrix multiplication is not possible." << endl;
        return 0;
    }
    int matrix1[rows1][cols1];
    int matrix2[rows2][cols2];
    int result[rows1][cols2];
    cout << "Enter elements of Matrix 1: " << endl;
    for (int i = 0; i < rows1; i++) {
        for (int j = 0; j < cols1; j++) {
            cin >> matrix1[i][j];
        }
    }
    cout << "Enter elements of Matrix 2: " << endl;
    for (int i = 0; i < rows2; i++) {
        for (int j = 0; j < cols2; j++) {
            cin >> matrix2[i][j];
```

```
    }
}
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];
        }
    }
}
cout << "Matrix 1: " << endl;
for (int i = 0; i < rows1; i++) {
    for (int j = 0; j < cols1; j++) {
        cout << matrix1[i][j] << " ";
    }
    cout << endl;
}
cout << "Matrix 2: " << endl;
for (int i = 0; i < rows2; i++) {
```

```
        for (int j = 0; j < cols2; j++) {
            cout << matrix2[i][j] << " ";
        }
        cout << endl;
    }
    cout << "Result of Matrix Multiplication: " << endl;
    for (int i = 0; i < rows1; i++) {
        for (int j = 0; j < cols2; j++) {
            cout << result[i][j] << " ";
        }
        cout << endl;
    }
    return 0;
}
```

Q23. WAP to sort an array in ascending order.

```
#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];
}
cout << "Original array: ";
for (int i = 0; i < n; i++) {
    cout << arr[i] << " ";
}
cout << endl;
for (int i = 0; i < n; i++) {
    for (int j = i + 1; j < n; j++) {
        if (arr[i] > arr[j]) {
            int temp = arr[i];
            arr[i] = arr[j];
            arr[j] = temp;
        }
```

```
        }
    }
    cout << "Sorted array: ";
    for (int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }
    cout << endl;
    return 0;
}
```

Q24. Write a program to reverse a given string.

```
#include <iostream>
using namespace std;
int main() {
string str, rev;
cout << “Enter a string : ”;
```

```
cin>> str;
for ( int i= str.length()-1; i >=0; i --)
{
 rev+=str[i];
}
cout<< "\n Revsersed string : "<<rev;
return0;
}
```


Q25. Write a program to count all the vowels in a given string.

```
#include <iostream>
using namespace std;
int main()
{
```

```
    string str;
    cout << "Enter a string: ";
    cin >> str;
    int vowelCount = 0;
    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')
         {
            vowelCount++;
          }
    }
    cout << "Number of vowels: " << vowelCount << endl;
    return 0;
}
```

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

```
#include<iostream>
using namespace std;
int main()
string st:
cout<<"Enter a string \n";
cin>>st:
int flag=0;
int len=st.size();
for (int i=0;i<len/2;i++)
if(st[i]! = st[len-1-i])
{
flag=1;
}
if(flag==0)
cout<<"Palindrome Word":
else
cout<<" Not Palindrome Word":
```

```
return 0;
```

Q27. WAP to check if a given string is anagram or not.

```
#include<iostream>

using namespace std;

int main(){
    int arr[26]={0};
    cout << "enter a size:";
    int size;
    cin >> size;
    cout << "s1:";
    char s1[size];
    for(int i=0;i<size;i++){
        cin >> s1[i];
```

```
}
char s2[size];
cout << "s2:";
for(int i=0;i<size;i++){
    cin >> s2[i];

}
for(int i=0;i<size;i++){
    int a = s1[i]-'a';
    arr[a]=arr[a]+1;
}
for(int i=0;i<size;i++){
    int a = s2[i]-'a';
    arr[a]=arr[a]-1;;
    // cout << a << endl;
}
int flag=0;
for(int i=0;i<26;i++){
    if(arr[i]!=0){
```

```
            flag=1;
            break;
        }
    }

    (flag==0)? cout << "true" : cout << "false";
    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{
```

```
    string make;
    string model;
    int year;
    public:
    void setData(){
        cout << "enter make:";
        cin >> make;
        cout << "enter model:";
        cin >> model;
        cout << "year:";
        cin >> year;
    }

    void getData(){
        cout << "make:" << make << endl;
        cout << "model:" << model << endl;
        cout << "year:" << year << endl;
    }
};
```

```
int main(){
    car c1;
    c1.setData();
    c1.getData();
    return 0;
}
```

Q29. Define a class called Address with attributes such as street, city, and zip Code. 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;
    Address(string s, string c, string z) {
        street = s;
        city = c;
        zipCode = z;
    }
};

class Person {
public:
    string name;
    Address address;
    Person(string n, string s, string c, string z) :
address(s, c, z) {
        name = n;
```

```
    }

    // Function to display Person details
    void displayDetails() {
        cout << "Name: " << name << endl;
        cout << "Address: " << address.street << ", " <<
address.city << " " << address.zipCode << endl;
    }
};

int main() {
    Person person("John Doe", "123 Main St", "Anytown",
"12345");
    person.displayDetails();
    return 0;
}
```

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];
    }
    // Find minimum
    int minVal = arr[0];
    for (int i = 1; i < n; i++) {
        if (arr[i] < minVal) {
            minVal = arr[i];
        }
    }
    // Find maximum
    int maxVal = arr[0];
```

```
for (int i = 1; i < n; i++) {
    if (arr[i] > maxVal) {
        maxVal = arr[i];
    }
}
// Calculate sum
int sum = 0;
for (int i = 0; i < n; i++) {
    sum += arr[i];
}
// Search for an element
int searchVal;
cout << "Enter the value to search: ";
cin >> searchVal;
bool found = false;
for (int i = 0; i < n; i++) {
    if (arr[i] == searchVal) {
        found = true;
        break;
```

```
        }
    }
    if (found) {
        cout << "Value found in the array." << endl;
    } else {
        cout << "Value not found in the array." << endl;
    }
    // Calculate average
    double average = (double)sum / n;

    // Display results
    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 {
private:
    int admno;
    char sname[20];
    float eng, math, science;
```

```
    float total;
public:
    // Function to calculate total
    float ctotal() {
        total = eng + math + science;
        return total;
    }

    void takeData() {
        cout << "Enter admission number: ";
        cin >> admno;
        cout << "Enter student name: ";
        cin >> sname;
        cout << "Enter English marks: ";
        cin >> eng;
        cout << "Enter Math marks: ";
        cin >> math;
        cout << "Enter Science marks: ";
        cin >> science;
```

```
    }
    void showData() {
        cout << "Admission Number: " << admno << endl;
        cout << "Student Name: " << sname << endl;
        cout << "English Marks: " << eng << endl;
        cout << "Math Marks: " << math << endl;
        cout << "Science Marks: " << science << endl;
        cout << "Total Marks: " << ctotal() << endl;
    }
};
int main() {
    Student student;
    student.takeData();
    student.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

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 travel{
    int flightNumber;
    string destination;
    int distance;
    float fuel;
    void calFuel(){
        if(distance<=1000) fuel=500;
        else if(distance>1000 && distance<=2000) fuel=1100;
        else fuel=2200;
    }
    public :
    void feedInfo(int fn,string des,int dist){
        flightNumber=fn;
        destination=des;
        distance=dist;
        calFuel();
    }
    void showInfo(){
        cout << "flight number:" << flightNumber << endl;
```

```
        cout << "destination:" << destination << endl;
        cout << "distance:" << distance << endl;
        cout << "fuel:" << fuel << endl;
    }
};

int main(){
     travel t1;
    t1.feedInfo(267,"indore",1100);
    t1.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;
class matrix{
    int arr1[3][3];
    int arr2[3][3];
    public :
    void Switch(int button){
        switch (button){
            case 1 :
                inputdata();
                break;
            case 2 :
                displaydata();
                break;
            case 3 :
                add();
                break;
            case 4:
```

```
            multiply();
            break;
        case 5:
            transpose();
            break;
        default:
            printf("Default case is Matched.");
            break;
    }
}
void inputdata(){
    cout << "array 1:";
    for(int i=0;i<3;i++){
        for(int j=0;j<3;j++) {
            cin >> arr1[i][j];
        }
    }
    cout << "array 2:";
    for(int i=0;i<3;i++){
```

```
        for(int j=0;j<3;j++){
         cin >> arr2[i][j];
        }
    }
}
void displaydata(){
    cout << "array 1:";
    for(int i=0;i<3;i++){
        for(int j=0;j<3;j++) {
            cout << arr1[i][j] << " ";
        }
        cout << endl;
    }
    cout << "array 2:";
    for(int i=0;i<3;i++){
        for(int j=0;j<3;j++) {
            cout << arr2[i][j] << " ";
        }
        cout << endl;
```

```
    }
}

void add(){
    cout << "sum of two matrix";
    for(int i=0;i<3;i++){
        for(int j=0;j<3;j++){
            int r= arr1[i][j]+arr2[i][j];
            cout << r << " ";
        }
        cout << endl;
    }
}

void multiply(){
    printf("the resultant matrix\n");
    for(int i=0;i<3;i++){
        int d=0;
        for(int j=0;j<i;j++){
```

```
            d=arr2[i][j];
            arr2[i][j]=arr2[j][i];
            arr2[j][i]=d;

        }
    }
    int r=0;
    for(int i=0;i<3;i++){
        for(int j=0;j<3;j++){
            for(int k=0;k<3;k++){
                r=arr1[i][k]*arr2[j][k]+r;
            }
            cout << r << " ";
        }
        cout << endl;
    }
}
void transpose(){
    cout << "transpose of both matrix:";
```

```
        for(int i=0;i<3;i++){
            int d=0;
            for(int j=0;j<i;j++){
                d=arr1[i][j];
                arr1[i][j]=arr1[j][i];
                arr1[j][i]=d;
            }
        }
        for(int i=0;i<3;i++){
            int d=0;
            for(int j=0;j<i;j++){
                d=arr2[i][j];
                arr2[i][j]=arr2[j][i];
                arr2[j][i]=d;
            }
        }
        displaydata();
    }
};
```

```
int main(){
    matrix m1;
    cout << "enter 1 for input matrix:" << endl;
    cout << "enter 2 for output matrix:" << endl;
    cout << "enter 3 for add two matrix:" << endl;
    cout << "enter 4 for multiply two matrix:" << endl;
    cout << "enter 5 for transpose of matrix:" << endl;
    int button;
    cout << "enter button:";
    cin >> button;
    m1.Switch(button);
    cout << "enter button:";
    cin >> button;
    m1.Switch(button);
    cout << "enter button:";
    cin >> button;
    m1.Switch(button);
    return 0;
```

```
}
```

--- END OF ASSIGNMENT---