C++ Programming Lab Assignment

1. WAP to find given no. is even or odd.

```cpp
//Even or Odd no.

#include<iostream>
using namespace std ;
int main() {
    int n;
    cout << "Enter a no. : ";
    cin >> n;
    if(n % 2 == 0){
        cout << n << " is an Even no." << endl;
    }
    else{
        cout << n << " is an Odd no." << endl;
    }
return 0;
}
```

Output :-
Enter a no. : 5
5 is Odd no.

2. WAP to find given no. is prime or composite.

```cpp
#include <iostream>

using namespace std;
//Prime or Composite no.
int main(){
    int n, temp=1;
    cout << "Enter a no. : " << endl;
    cin >> n;
    for(int i=2 ; i<=n-1 ; i++){
        if(n % i == 0){
            cout << n << " is a Composite no." << endl;
            temp=0;
            break;
        }
    }
    if(temp==1){
        cout << n << " is a Prime no." << endl;
    }
return 0;
```

```
}
```

Output :-

Enter a no.

40

40 is a Composite no.

PS C:\Users\DE\projects\Code++> g++ PrimeorComp.cpp

PS C:\Users\DE\projects\Code++> .\a.exe

Enter a no. :

23

23 is a Prime no.

3. WAP to print table of a given no. upto ‘n’ multiples.

```
#include<iostream>

using namespace std;
//table of a no.
int main(){
    int n, t, l;
    cout << "Enter values of n :" << endl;
    cin >> n ;
    cout << "Enter no. of terms : " << endl;
    cin >> l;
    cout << "Table of " << n << ": " << endl;
    for(int i=1 ; i<=l ; i++){
        t = n * i;
        cout << n << " x " << i << " = " << t << endl;
        }
return 0;
    }
```

Output :-

Enter values of n :

15

Enter no. of terms :

10

Table of 15:

15 x 1 = 15

15 x 2 = 30

15 x 3 = 45

15 x 4 = 60

15 x 5 = 75

15 x 6 = 90
15 x 7 = 105
15 x 8 = 120
15 x 9 = 135
15 x 10 = 150

4. WAP to find i) greater of two no. ii) greatest of three no.

```
#include <iostream>

using namespace std;
//greatest no. among the two no.
int main(){
    int a, b;
    cout << "Enter two no. :" << endl;
    cin >> a >> b ;
    (a>b) ? cout << a << " is greatest" : cout << b << " is greatest" << endl;

//greatest no. among the three no.
    int x, y, z;
    cout << "\nEnter three no. :" << endl;
    cin >> x >> y >> z ;
    (x>y) ? (x>z) ? cout << x << " is greatest" : cout << z << " is greatest"
: (y>z) ? cout << y << " is greatest" : cout << z << " is greatest" << endl;
return 0;
}
```

Output :-
Enter two no. :
25
56
56 is greatest

Enter three no. :
12
45
5
45 is greatest

5. WAP to find sum of first ‘n’ natural no.

```
# include<iostream>

using namespace std;
//Sum of first n no.
int main(){
    int n, sum=0;
    cout << "Enter no. of terms :" << endl;
    cin >> n;
    for(int i=1 ; i<=n ; i++){
        sum = sum + i;
    }
    cout << "Sum of first " << n << " no. is : " << sum << endl;
return 0;
}
```

Output :-
Enter no. of terms :
10
Sum of first 10 no. is : 55

6. WAP to find factorial of a given no.

```
//Factorial of a no.

#include<iostream>
using namespace std;
int main(){
    int a, fact = 1;
    cout << "Enter a no. :" << endl;
    cin >> a;
    cout << "Factorial of " << a << " is :" << endl;
    for( int k=1 ; k<=a ; k++){
        fact = fact * k;
    }
    cout << fact << endl;
return 0;
}
```

Output :-
Enter a no. :
5
Factorial of 5 is :
120

7. WAP to find sum of digits of ‘n’ digit no.

```
# include<iostream>

using namespace std;
//Sum of digits of a no.
int main(){
    int a, result=0, rem=0;
    cout << "Enter any no. :" << endl;
    cin >> a;
    while(a>0){
        rem = a % 10;
        result = result + rem;
        a = a/10;
    }
    cout << "Sum of all digits is = " << result << endl;
return 0;
}
```

Output :-
Enter any no. :
435
Sum of all digits is = 12

8. WAP to find reverse of a no.
9. Check whether the given no. is palindrome or not.

```
#include <iostream>

using namespace std;
//reverse of a no. or Palindrome
int main(){
    int n,revnum=0, rem=0, originalnum;
    cout << "Enter any no. :" << endl;
    cin >> n;
    originalnum = n;
    while(originalnum > 0){
        rem = originalnum % 10;
        revnum = revnum * 10 + rem;
        originalnum = originalnum / 10;
    }
    cout << "Reverse of no. " << n << " is : " << revnum << endl;
    if(n == revnum){
        cout << "Entered no. is a Palindrome no." << endl;
    }
    else{
        cout << "Entered no. is not a Palindrome no." << endl;
    }
return 0;
```

```
}
```

Output :-
Enter any no. :
234
Reverse of no. 234 is : 432
Entered no. is not a Palindrome no.
PS C:\Users\DE\projects\Code++> g++ RevPal.cpp
PS C:\Users\DE\projects\Code++> .\a.exe
Enter any no. :
43534
Reverse of no. 43534 is : 43534
Entered no. is a Palindrome no.

10. WAP to print Fibonacci series upto ‘n’ terms.

```cpp
#include <iostream>

using namespace std;
//Fibonacci series upto n terms
int main(){
    int n, a=0, b=1, temp;
    cout << "Enter no. of terms : " << endl;
    cin >> n;
    cout << "0 " << "1 ";
    for(int i=1 ; i<=n-2 ; i++){
        temp = a + b;
        cout << temp << " ";
        a = b;
        b = temp;
    }
return 0;
}
```

Output :-
Enter no. of terms :
10
0 1 1 2 3 5 8 13 21 34

11. WAP to determine given ‘n’ digit no. is Armstrong no. or not.

```cpp
#include <iostream>
using namespace std;
//Armstrong no. for a 'n' digit no.
int main(){
    int num, originalnum, rem=0, result=0, n=0;
    cout << "Enter any no. of digit no. :" << endl;
    cin >> num;
    originalnum = num;

    for(int originalnum = num; originalnum !=0; n++){
        originalnum /= 10;
    }

    originalnum = num;

    for(int originalnum = num; originalnum !=0; originalnum /= 10){
        rem = originalnum % 10;
        int power = 1;   //for sum of powers
        for(int i=0; i<n; i++){
            power = power * rem;
        }
        result = result + power;
    }
    cout << "Result = " << result << endl;
    if(num == result){
        cout << num << " is a Armstrong no." << endl;
    }
    else{
        cout << num << " is not a Armstrong no." << endl;
    }
return 0;
}
```

Output :-

```
Enter any no. of digit no. :
153
Result = 153
153 is a Armstrong no.
PS C:\Users\DE\projects\Code++> g++ ArmsorFib.cpp
PS C:\Users\DE\projects\Code++> .\a.exe
Enter any no. of digit no. :
4567
Result = 4578
4567 is not a Armstrong no.
```

12. WAP to print all even no. between 100 & 200.

```cpp
#include <iostream>
using namespace std;
//to print all even no. between 100 & 200
int main(){
    cout << "All the even no. between 100 & 200 are : " << endl;
    for(int i=100 ; i<=200 ; i++){
        if(i % 2 == 0){
            cout << i << " ";
        }
    }
return 0;
}
```

Output :-
All the even no. between 100 & 200 are :
100 102 104 106 108 110 112 114 116 118 120 122 124 126 128 130 132 134 136 138 140 142 144 146 148 150 152 154 156 158 160 162 164 166 168 170 172 174 176 178 180 182 184 186 188 190 192 194 196 198 200

13. WAP to print first 50 prime no.

```cpp
 #include <iostream>

using namespace std;
//to print first 50 prime no.
int main(){
    cout << "\nFirst 50 Prime no. are : " << endl;
    int count = 0;
    for (int num = 2; ; ++num) {
        bool isPrime = true;
        for (int j = 2; j <= num-1; ++j) {
            if (num % j == 0) {
                isPrime = false;
                break;
            }
        }
        if (isPrime) {
            cout << num << " ";
            ++count;
        }
        if (count == 50) {
            break;
        }
    }
    return 0;
}
```

Output :-
First 50 Prime no. are :
3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89
97 101 103 107 109 113 127 131 137 139 149 151 157 163 167 173
179 181 191 193 197 199 211 223 227 229

14. WAP to print all 4 digits Armstrong no.

```
#include <iostream>
using namespace std;
//to print all four digit armstrong no.
int main(){
int num, originalnum, rem=0, result=0;
    for(int num = 1000 ; num < 10000 ; num++){
        originalnum = num;
        result = 0;

    while(originalnum > 0){
        rem = originalnum % 10;
        result = result + rem * rem * rem * rem;
        originalnum = originalnum / 10;
    }

    if(num == result){
        cout << num << endl;
    }
    }

return 0;
}
```

Output :-
All 4 digits Armstrong no. are :
1634
8208
9474

15. WAP to print the following patterns :

```
/*to print *     or     * * * * *    or       *    or
           * *          * * * *             * * *
           * * *        * * *             * * * * *
           * * * *      * *             * * * * * * *
           * * * * *    *             * * * * * * * * *    */
#include <iostream>
using namespace std;
int main(){
    int n;
    cout << "Enter no. of rows :" << endl;
    cin >> n;
    cout << "Star pattern in increasing order :" << endl;
    for(int i=0 ; i<n ; i++){
        for(int j=0 ; j<=i ; j++){
            cout << "* ";
        }
        cout << endl;
    }
    cout << "Star patter in decreasing order :" << endl;
    for(int k=n ; k>0 ; k--){
        for(int l=0 ; l<k ; l++){
            cout << "* ";
        }
        cout << endl;
    }
    cout << "Pyramid star pattern :" << endl;
    for (int a = 1; a <= n; a++) {
        for (int b = a; b < n; b++) {
            cout << "  "; // Print spaces
        }
        for (int c = 1; c <= (2 * a - 1); c++) {
            cout << "* "; // Print stars
        }
        cout << endl;
    }
return 0;
}
```

Output :-

Enter no. of rows :

5

Star pattern in increasing order :

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

```
* * * * *
```

Star patter in decreasing order :

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

Pyramid star pattern :

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

```
/*to print 1        or      1     or   1
           2 2             1 1         2 3
           3 3 3          1 2 1        4 5 6
           4 4 4 4       1 3 3 1       7 8 9 10
           5 5 5 5 5    1 4 6 4 1      11 12 13 14 15 */
#include <iostream>
using namespace std;
int main(){
    int n;
    cout << "Enter no. of rows :" << endl;
    cin >> n;
    cout << "\nNo. pattern in increasing order :" << endl;
    for(int i=1 ; i<=n ; i++){
        for(int j=1 ; j<=i ; j++){
            cout << i << " " ;
        }
        cout << endl;
    }
    cout << "\nPascal's triangle :" << endl;
    for (int line = 0; line < n; line++) {

        //for spaces
        for (int space = 0; space < n - line - 1; space++) {
            cout << "  ";
        }
        int number = 1; // Start each line with 1
        for (int j = 0; j <= line; j++) {
            cout << number << "   ";
            number = number * (line - j) / (j + 1); // Calculate the next
number in the line
        }
        cout << endl;
    }
```

```
    cout << "\nfloyd's triangle :" << endl;
    int num=1;
    for(int a=0 ; a<n ; a++){
        for(int b=a+1 ; b>0 ; b--){
            cout << num << " ";
            num++;
        }
        cout << endl;
    }
return 0;
}
```

Output :-
Enter no. of rows :
5
No. pattern in increasing order :
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5

Pascal's triangle :
```
      1
    1   1
  1   2   1
 1   3   3   1
1   4   6   4   1
```

Floyd’s triangle :
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15

16. Using Functions, write following c++ programs :

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

```
#include <iostream>

using namespace std;
//to print all palindrome between 500 & 1000
int isPalindrome(int n){
    int revnum=0, rem=0, originalnum;
    originalnum = n;
    while(originalnum > 0){
        rem = originalnum % 10;
        revnum = revnum * 10 + rem;
        originalnum = originalnum / 10;
    }
    return (n == revnum);
    }
    int main(){
    cout << "All Palindromes between 500 & 1000 are : " << endl;
    for(int i=500; i<=1000; i++){
         if (isPalindrome(i)){
            cout << i << endl;
         }
    }
    return 0;
    }
```

Output :-

All Palindromes between 500 & 1000 are :

505 515 525 535 545 555 565 575 585 595 606 616 626 636 646 656 666 676 686 696 707 717 727 737 747 757 767 777 787 797 808 818 828 838 848 858 868 878 888 898 909 919 929 939 949 959 969 979 989 999

ii) To print first 100 odd no.

```
#include <iostream>
using namespace std;
//to print first 100 odd no.
int isOdd(int n){
    return (n % 2 != 0);
    }
int main(){
    cout << "First 100 odd no. are : " << endl;
    int count = 0; // Counter for the number of odd numbers printed
    int i = 1; // Start with the first odd number
    while (count < 100) {
        if (isOdd(i)) {
            cout << i << " ";
```

```
            count++;
        }
        i++;
    }
    return 0;
}
```

Output :-

First 100 odd no. are :

1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55 57 59 61 63 65 67 69 71 73 75 77 79 81 83 85 87 89 91 93 95 97 99 101 103 105 107 109 111 113 115 117 119 121 123 125 127 129 131 133 135 137 139 141 143 145 147 149 151 153 155 157 159 161 163 165 167 169 171 173 175 177 179 181 183 185 187 189 191 193 195 197 199

iii) To find binary, octal & hexadecimal equivalent of a given decimal no.

```
#include <iostream>
using namespace std;
//to find binary, octal & hexadecimal equivalent of a given decimal no.
int DecToBinary(int decNum){
    int ans=0, pow=1;
    while(decNum>0){
        int rem = decNum % 2;
        decNum = decNum / 2;
        ans += rem * pow;
        pow = pow * 10;
    }
    return ans;
}
int DecToOctal(int decNum){
    int ans=0, pow=1;
    while(decNum>0){
        int rem = decNum % 8;
        decNum = decNum / 8;
        ans += rem * pow;
        pow = pow * 10;
    }
    return ans;
}
int DecTohex(int decNum){
    int ans=0, pow=1;
    while(decNum>0){
        int rem = decNum % 16;
        decNum = decNum / 16;
        ans += rem * pow;
```

```
        pow = pow * 16;
    }
    return ans;
}
int decimalTohex(int decNum) {
    cout << "Hexadecimal equivalent: " << hex << decNum << endl;
}
int main(){
    int decNum;
    cout << "Enter a decimal no. : " << endl;
    cin >> decNum;
    cout << "Binary equivalent of " << decNum << " is : " <<
DecToBinary(decNum) << endl;
    cout << "Octal equivalent of " << decNum << " is : " << DecToOctal(decNum)
<< endl;
    cout << "Hexadecimal equivalent of " << decNum << " is : " <<
DecTohex(decNum) << endl;
    cout << decimalTohex(decNum) << endl;
return 0;
}
```

Output :-
Enter a decimal no. :
12
Binary equivalent of 12 is : 1100
Octal equivalent of 12 is : 14
Hexadecimal equivalent of 12 is : 12
Hexadecimal equivalent: c

iv) To find decimal equivalent for given binary, octal & hexadecimal no.

```
#include <iostream>
using namespace std;
//to find decimal equivalent for given binary, octal & hexadecimal no.
int binaryToDec(int biNum){
    int ans=0, pow=1;
    while(biNum>0){
        int rem = biNum % 10;
        biNum = biNum / 10;
        ans += rem * pow;
        pow = pow * 2;
    }
    return ans;
}
int octalToDec(int octNum){
    int ans=0, pow=1;
```

```
    while(octNum>0){
        int rem = octNum % 10;
        octNum = octNum / 10;
        ans += rem * pow;
        pow = pow * 8;
    }
    return ans;
}
int hexToDec(int hexNum){
    int ans=0, pow=1;
    while(hexNum>0){
        int rem = hexNum % 10;
        hexNum = hexNum / 10;
        ans += rem * pow;
        pow = pow * 16;
    }
    return ans;
}
int main(){
    int biNum, octNum, hexNum;
    cout << "Decimal equivalent of a given binary no. is : " <<
binaryToDec(1010) << endl;
    cout << "Decimal equivalent of a given octal no. is :" << octalToDec(1010)
<< endl;
    cout << "Decimal equivalent of a given hexadecimal no. is : " <<
hexToDec(1010) << endl;
    return 0;
}
```

Output :-
Decimal equivalent of a given binary no. is : 10
Decimal equivalent of a given octal no. is :520
Decimal equivalent of a given hexadecimal no. is : 4112

v) To calculate geometric sum upto ‘n’ terms.

```
#include <iostream>
using namespace std;
//to find geometric sum upto n terms
int GeometricSum(int n, int r){
    int sum=0, term=1;
    for(int i=0; i<n; i++){
        sum = sum + term;
        term = term * r;
    }
    return sum;
}
int main(){
```

```
    int n, r;
    cout << "Enter the no. of terms(n) : " << endl;
    cin >> n;
    cout << "Enter the common ratio(r) : " << endl;
    cin >> r;
    cout << "The geometric sum of " << n << " terms with a common ratio of "
<< r << " is : " << GeometricSum(n,r) << endl;
return 0;
}
```

Output :-

Enter the no. of terms(n) :

5

Enter the common ratio(r) :

2

The geometric sum of 5 terms with a common ratio of 2 is : 31

17. Using recursive function, write following c++ programs :
    i) To print binary no. for a given decimal no.
    ii) To print octal no. for a given decimal no.

```
#include <iostream>
using namespace std;
//to print binary & octal no. for a given decimal no.
void decTobinary(int decNum){
    if(decNum > 1){
        decTobinary(decNum / 2);
    }
    cout << decNum % 2;
}
void decTooctal(int decNum){
    if(decNum > 7){
        decTooctal(decNum / 8);
    }
    cout << decNum % 8;
}
int main(){
    int decNum;
    cout << "Enter a decimal no. " << endl;
    cin >> decNum;
    cout << "Binary no. for " << decNum << " is : ";
    decTobinary(decNum);
    cout << endl;
    cout << "Octal no. for " << decNum << " is : ";
    decTooctal(decNum);
    cout << endl;
return 0;
}
```

Output :-
Enter a decimal no.
15
Binary no. for 15 is : 1111
Octal no. for 15 is : 17

iii) To print factorials for a given range.

```cpp
#include <iostream>
using namespace std;
//to print factorials for a given range
int fact(int a){
    if(a <= 1){
        return 1;
    }
    else{
        return (a * fact(a-1));
    }
}
int main(){
    int start, end;
    cout << "Enter the start of the range : " << endl;
    cin >> start;
    cout << "Enter the end of the range : " << endl;
    cin >> end;
    for(int i = start; i <= end; i++){
        int f = fact(i);
        cout << "Factorial of " << i << " is " << f << endl;
    }
return 0;
}
```

Output :-
Enter the start of the range :
1
Enter the end of the range :
5
Factorial of 1 is 1
Factorial of 2 is 2
Factorial of 3 is 6
Factorial of 4 is 24
Factorial of 5 is 120

iv) To print first ‘n’ terms of Fibonacci series.

```
#include <iostream>
using namespace std;
//to print first ‘n’ terms of Fibonacci series
int fibonacci(int n){
    if(n <= 1){
        return n;
    }
    else{
        return (fibonacci(n-1) + fibonacci(n-2));
    }
}
int main(){
    int n;
    cout << "Enter no. of terms : " << endl;
    cin >> n;
    cout << "Fibonacci series upto " << n << " terms : " << endl;
    for(int i=0; i<n; i++){
        cout << fibonacci(i) << " ";
    }
    return 0;
}
```

Output :-
Enter no. of terms :
10
Fibonacci series upto 10 terms :
0 1 1 2 3 5 8 13 21 34

18. WAP to calculate average of all elements of a 1D array.
19. WAP to find out minimum & maximum value of a 1D numeric array.

```
#include <iostream>
using namespace std;
//to print sum & avg of an array and max & min no.
int main(){
    int a[10], sum=0, l=0, s;
    float avg=0;
    cout << "Enter 10 no.: " << endl;
    for(int i=0; i<=9; i++){
        cin >> a[i];
    }
    l = a[0]; s = a[0];
    for(int i=0; i<=9; i++){
        sum = sum + a[i];
```

```
        if(l < a[i])
        l = a[i];
        if(s > a[i])
        s = a[i];
    }
    cout << "Sum of entered no.= " << sum << endl;
    avg = (float)sum/10;
    cout << "Average of entered no.= " << avg << endl;
    cout << "Maximum no.= " << l << endl;
    cout << "Minimum no.= " << s << endl;
return 0;
}
```

Output :-
Enter 10 no.:
1
2
3
4
5
6
7
8
9
10
Sum of entered no.= 55
Average of entered no.= 5.5
Maximum no.= 10
Minimum no.= 1

20. WAP to find transpose of a 2D matrix.

```
#include <iostream>
using namespace std;
//transpose of a matrix or 2D array
int main(){
    int a[3][3];
    cout << "Enter matrix elements : " << endl;
    for(int i=0; i<=2; i++){
        for(int j=0; j<=2; j++){
            cin >> a[i][j];
        }
    }
    cout << "Entered matrix : " << endl;
    for(int i=0; i<=2; i++){
        for(int j=0; j<=2; j++){
```

```
            cout << a[i][j] << " ";
        }
        cout << "\n";
    }
    cout << "Transpose of entered matrix : " << endl;
    for(int i=0; i<=2; i++){
        for(int j=0; j<=2; j++){
            cout << a[j][i] << " ";
        }
        cout << "\n";
    }
return 0;
}
```

Output :-

```
Enter matrix elements :
1
2
3
4
5
6
7
8
9
Entered matrix :
1 2 3
4 5 6
7 8 9
Transpose of entered matrix :
1 4 7
2 5 8
3 6 9
```

21. WAP to add 2D matrices.

```cpp
#include <iostream>
using namespace std;
//to add 2 matrices
int main(){
    int rows, columns;
    cout << "Enter the no. of Rows & Columns : " << endl;
    cin >> rows >> columns;
    int M1[rows][columns], M2[rows][columns], sumM[rows][columns];
    cout << "Enter matrix1 elements : " << endl;
    for(int i=0; i<rows; i++){
        for(int j=0; j<columns; j++){
            cin >> M1[i][j];
        }
    }
    cout << "Matrix1 : " << endl;
    for(int i=0; i<rows; i++){
        for(int j=0; j<columns; j++){
            cout << M1[i][j] << " ";
        }
        cout << "\n";
    }
    cout << "Enter matrix2 elements : " << endl;
    for(int i=0; i<rows; i++){
        for(int j=0; j<columns; j++){
            cin >> M2[i][j];
        }
    }
    cout << "Matrix2 : " << endl;
    for(int i=0; i<rows; i++){
        for(int j=0; j<columns; j++){
            cout << M2[i][j] << " ";
        }
        cout << "\n";
    }

    for(int i=0; i<rows; i++){
        for(int j=0; j<columns; j++){
            sumM[i][j] = M1[i][j] + M2[i][j];
        }
    }
    cout << "Sum of entered matrices : " << endl;
    for(int i=0; i<rows; i++){
        for(int j=0; j<columns; j++){
        cout << sumM[i][j] << " ";
        }
        cout << endl;
    }
```

```
return 0;
}
```

Output :-

```
Enter the no. of Rows & Columns :
2
3
Enter matrix1 elements :
1
2
3
4
5
6
Matrix1 :
1 2 3
4 5 6
Enter matrix2 elements :
7
8
9
1
2
3
Matrix2 :
7 8 9
1 2 3
Sum of entered matrices :
8 10 12
5 7 9
```

22. WAP to multiply 2D matrices.

```cpp
#include <iostream>
using namespace std;
//to multiply 2 matrices
int main(){
    int rows1, columns1, rows2, columns2;
    cout << "Enter the no. of Rows & Columns for matrix1 : " << endl;
    cin >> rows1 >> columns1;
    cout << "Enter the no. of Rows & Columns for matrix2 : " << endl;
    cin >> rows2 >> columns2;
    if(columns1 != rows2){
        cout << "Matrices cannot be multiplied." << endl;
        return 0;
    }
    int M1[rows1][columns1], M2[rows2][columns2], productM[rows1][columns2];
    cout << "Enter matrix1 elements : " << endl;
    for(int i=0; i<rows1; i++){
        for(int j=0; j<columns1; j++){
            cin >> M1[i][j];
        }
    }
    cout << "Matrix1 : " << endl;
    for(int i=0; i<rows1; i++){
        for(int j=0; j<columns1; j++){
            cout << M1[i][j] << " ";
        }
        cout << "\n";
    }
    cout << "Enter matrix2 elements : " << endl;
    for(int i=0; i<rows2; i++){
        for(int j=0; j<columns2; j++){
            cin >> M2[i][j];
        }
    }
    cout << "Matrix2 : " << endl;
    for(int i=0; i<rows2; i++){
        for(int j=0; j<columns2; j++){
            cout << M2[i][j] << " ";
        }
        cout << "\n";
    }

    for (int i = 0; i<rows1; i++) {
        for (int j=0; j<columns2; j++) {
            productM[i][j] = 0;
            for (int k=0; k<columns1; k++) {
                productM[i][j] += M1[i][k] * M2[j][k];
            }
```

```
        }
    }
    cout << "Product of entered matrices : " << endl;
    for (int i=0; i<rows1; i++) {
        for (int j=0; j<columns2; j++) {
            cout << productM[i][j] << " ";
        }
        cout << endl;
    }
return 0;
}
```

Output :-

```
Enter the no. of Rows & Columns for matrix1 :
2
3
Enter the no. of Rows & Columns for matrix2 :
2
4
Matrices cannot be multiplied.
PS C:\Users\DE\projects\Code++> g++ MultiplyM.cpp
PS C:\Users\DE\projects\Code++> .\a.exe
Enter the no. of Rows & Columns for matrix1 :
2
3
Enter the no. of Rows & Columns for matrix2 :
3
2
Enter matrix1 elements :
1
2
3
4
5
6
Matrix1 :
1 2 3
4 5 6
Enter matrix2 elements :
7
8
9
```

0
1
3
Matrix2 :
7 8
9 0
1 3
Product of entered matrices :
50 12
122 42

23. WAP to sort an array in ascending order.

```cpp
#include <iostream>
using namespace std;
//to sort an array in ascending order
int main(){
    int n;
    cout << "Enter the no. of elements in the array : ";
    cin >> n;
    int arr[n];
    cout << "Enter the elements of the array:" << endl;
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }

    // Sort the array
    for (int i = 0; i < n-1; i++) {
        for (int j = i+1; j < n; j++) {
            if (arr[i] > arr[j]) {
                // Swap arr[i] and arr[j]
                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }

    // Display the sorted array
    cout << "Sorted array in ascending order : " << endl;
    for (int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }
    cout << endl;
return 0;
}
```

Output :-
Enter the no. of elements in the array : 5
Enter the elements of the array:
12
34
56
10
5
Sorted array in ascending order :
5 10 12 34 56

24. WAP to reverse a given string.
25. WAP to check if a given string is palindrome or not.

```cpp
#include <iostream>
#include <algorithm>
#include <string>
using namespace std;
//to reverse a given string & check if it is a Palindrome or not
int main(){
    string str, revstr;
    cout << "Enter a string : " << endl;
    cin >> str;
    for(int i = str.length()-1; i>=0; i--){
       revstr = revstr + str[i];
    }
    cout << "Reversed String : " << revstr << endl;
    if(str == revstr){
        cout << "String is a Palindrome." << endl;
    }
    else{
        cout << "String is not a Palindrome." << endl;
    }
return 0;
}
```

Output :-
Enter a string :
Programming
Reversed String : gnimmargorP
String is not a Palindrome.
PS C:\Users\DE\projects\Code++> g++ RevString.cpp
PS C:\Users\DE\projects\Code++> .\a.exe
Enter a string :
naman
Reversed String : naman
String is a Palindrome.

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

```
#include <iostream>
#include <string.h>
using namespace std;
//to count all the vowels in a given string
int main(){
    string str; char c;
    int vowelCount=0;
    cout << "Enter a string : " << endl;
    cin >> str;
    cout << "Vowels in the given string are : " << endl;
    for(char c : str){
        if(c=='a' || c=='e' || c=='i' || c=='o' || c=='u' ||
           c=='A' || c=='E' || c=='I' || c=='O' || c=='U'){
            cout << c << " ";
            vowelCount++;
            }
    }
    cout << "\nNo. of vowels in the string are : " << vowelCount << endl;
return 0;
}
```

Output :-
Enter a string :
education
Vowels in the given string are :
e u a i o
No. of vowels in the string are : 5

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

```
#include <iostream>
#include <algorithm>
#include <string.h>
using namespace std;
//to check if a given string is anagram or not
int main(){
    string str1, str2;
    cout << "Enter first string : " << endl;
    cin >> str1;
    cout << "Enter second string : " << endl;
    cin >> str2;
    // Check if the strings have the same length
    if(str1.length() != str2.length()){
        cout << "The strings are not anagrams." << endl;
        return 0;
    }
    // Sort the strings
```

```
    sort(str1.begin(), str1.end());
    sort(str2.begin(), str2.end());

    if(str1 == str2){
        cout << "The strings are anagrams." << endl;
    }
    else{
        cout << "The strings are not anagrams." << endl;
    }
return 0;
}
```

Output :-
Enter first string :
gum
Enter second string :
mug
The strings are anagrams.
PS C:\Users\DE\projects\Code++> g++ Anagrams.cpp
PS C:\Users\DE\projects\Code++> .\a.exe
Enter first string :
hello
Enter second string :
hey
The strings are not anagrams.

28. WAP to search an element of an array.

```
#include <iostream>
using namespace std;
//to search an element of an array
int main(){
    int n, search;
    cout << "Enter the size of the array : " << endl;
    cin >> n;
    int arr[n];
    cout << "Enter the elements of the array : " << endl;
    for(int i=0; i<n; i++){
        cin >> arr[i];
    }
    cout << "Enter the element to search : " << endl;
    cin >> search;
    for(int i=0; i<n; i++){
        if(arr[i] == search){
            cout << "Element found at index : " << i << endl;
            return 0;
```

```
        }
    }
    cout << "Element not found in the given array." << endl;
    return 0;
}
```

Output :-
Enter the size of the array :
5
Enter the elements of the array :
1
2
3
4
5
Enter the element to search :
7
Element not found in the given array.

29. Define a class student with the following specifications :

Private members of the 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, math, science.

showData() Function to display all the data members on the screen

```
#include <iostream>
#include <string.h>
using namespace std;
class student{
    private :
        int Admno;
        string Sname;
        float eng, math, science, Total;

    public :
        float ctotal(){
        Total = eng + math + science;
```

```
        return Total;
        }
    void takeData(int admno, const string sname, float e, float m, float sci){
        Admno = admno;
        Sname = sname;
        eng = e;
        math = m;
        science = sci;
        ctotal();
    }
    void showData() const{
        cout << "Admission No. : " << Admno << endl;
        cout << "Student name : " << Sname << endl;
        cout << "Marks of English : " << eng << endl;
        cout << "Marks of Maths : " << math << endl;
        cout << "Marks of Science : " << science << endl;
        cout << "Total marks : " << Total << endl;
    }
};
int main(){
    student s;
    s.takeData(1, "John", 95.0, 90.5, 92.0);
    s.showData();
return 0;
}
```

Output :-
Admission No. : 1
Student name : John
Marks of English : 95
Marks of Maths : 90.5
Marks of Science : 92
Total marks : 277.5

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

Private Members

A data member Flight no. 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 & <=2000 1100
More than 2000 2200

Public Members

A function FEEDINFO() to allow user to enter values for Flight no., 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.h>
using namespace std;
class Flight{
    private :
        int Flightno;
        string Destination;
        float Distance, Fuel;
    void CALFUEL(){
        if(Distance <= 1000){
            Fuel = 500;
        }
        else if(Distance > 1000 && Distance <= 2000){
            Fuel = 1100;
        }
        else{
            Fuel = 2200;
        }
    }
    public :
        void FEEDINFO(){
            cout << "Enter flight no. : " << endl;
            cin >> Flightno;
            cout << "Enter Destination : " << endl;
            cin >> Destination;
            cout << "Enter Distance : " << endl;
            cin >> Distance;
            CALFUEL();
        }
        void SHOWINFO(){
            cout << "Flight No. : " << Flightno << endl;
            cout << "Destination : " << Destination << endl;
            cout << "Distance : " << Distance << endl;
            cout << "Fuel Required : " << Fuel << endl;
        }
};
int main(){
    Flight;
    flight.FEEDINFO();
```

```
    flight.SHOWINFO();
return 0;
}
```

Output :-
Enter flight no. :
123
Enter Destination :
India
Enter Distance :
5000
Flight No. : 123
Destination : India
Distance : 5000
Fuel Required : 2200

31. Write a menu driven program to perform following :
    a) Input a matrix
    b) Display matrix
    c) Add two matrices
    d) Multiply two matrices
    e) Transpose a matrix

```
#include<iostream>
using namespace std;
void inputMatrix(int M[][20], int& row, int& col) {
    cout << "Enter the number of rows (1-20): ";
    cin >> row;
    cout << "Enter the number of columns (1-20): ";
    cin >> col;
    cout << "Enter the elements : ";
    for (int i = 0; i < row; ++i) {
        for (int j = 0; j < col; ++j) {
            cin >> M[i][j];
        }
    }
}

void outputMatrix(int M[][20], int row, int col) {
    for (int i = 0; i < row; ++i) {
        for (int j = 0; j < col; ++j) {
            cout << M[i][j] << " ";
        }
        cout << endl;
    }
}
```

```
void addMatrices(int M1[][20], int M2[][20], int sum[][20], int row, int col)
{
    for (int i = 0; i < row; ++i) {
        for (int j = 0; j < col; ++j) {
            sum[i][j] = M1[i][j] + M2[i][j];
        }
    }
    cout << "Sum of Matrices : " << endl;
    outputMatrix(sum, row, col);
}

void transpose(int M[][20], int row, int col) {
    cout << "Transpose of matrix: " << endl;
    for (int i = 0; i < col; ++i) {
        for (int j = 0; j < row; ++j) {
            cout << M[j][i] << " ";
        }
        cout << endl;
    }
}

void multiplyMatrices(int M1[][20], int M2[][20], int product[][20], int row1,
int col1, int row2, int col2) {
    if (col1 != row2) {
        cout << "Matrices cannot be multiplied." << endl;
        return;
    }
    for (int i = 0; i < row1; ++i) {
        for (int j = 0; j < col2; ++j) {
            product[i][j] = 0;
            for (int k = 0; k < col1; ++k) {
                product[i][j] += M1[i][k] * M2[k][j];
            }
        }
    }
    cout << "Product of Matrices: " << endl;
    outputMatrix(product, row1, col2);
}

int main() {
    int M1[20][20], M2[20][20], sum[20][20], product[20][20];
    int row1, row2, col1, col2, choice;

    do {
        cout << "\nMatrix Operations Menu :" << 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 (1-6): ";
        cin >> choice;

        switch (choice) {
            case 1: {
                cout << "For Matrix 1: " << endl;
                inputMatrix(M1, row1, col1);
                cout << "For Matrix 2: " << endl;
                inputMatrix(M2, row2, col2);
                break;
            }
            case 2: {
                cout << "Matrix 1:" << endl;
                outputMatrix(M1, row1, col1);
                cout << "Matrix 2:" << endl;
                outputMatrix(M2, row2, col2);
                break;
            }
            case 3: {
                if (row1 == row2 && col1 == col2) {
                    addMatrices(M1, M2, sum, row1, col1);
                } else {
                    cout << "Invalid Addition! Matrices dimensions do not
match." << endl;
                }
                break;
            }
            case 4: {
                multiplyMatrices(M1, M2, product, row1, col1, row2, col2);
                break;
            }
            case 5: {
                cout << "Transpose of Matrix 1:" << endl;
                transpose(M1, row1, col1);
                cout << "Transpose of Matrix 2:" << endl;
                transpose(M2, row2, col2);
                break;
            }
            case 6: {
                cout << "Exiting program." << endl;
                break;
            }
            default: {
                cout << "Invalid choice! Please try again." << endl;
```

```
            }
        }
    } while (choice != 6);

return 0;
}
```

Output :-
Matrix Operations Menu :
1. Input Matrix
2. Display Matrix
3. Add Two Matrices
4. Multiply Two Matrices
5. Transpose a Matrix
6. Exit
Enter your choice (1-6): 1
For Matrix 1:
Enter the number of rows (1-20): 2
Enter the number of columns (1-20): 2
Enter the elements: 1
2
3
4
For Matrix 2:
Enter the number of rows (1-20): 2
Enter the number of columns (1-20): 2
Enter the elements: 5
6
7
8

Matrix Operations Menu:
1. Input Matrix
2. Display Matrix
3. Add Two Matrices
4. Multiply Two Matrices
5. Transpose a Matrix
6. Exit
Enter your choice (1-6): 5
Transpose of Matrix 1:
Transpose of matrix:

```
1 3
2 4
Transpose of Matrix 2:
Transpose of matrix:
5 7
6 8

Matrix Operations Menu:
1. Input Matrix
2. Display Matrix
3. Add Two Matrices
4. Multiply Two Matrices
5. Transpose a Matrix
6. Exit
Enter your choice (1-6): 3
Sum of Matrices:
6 8
10 12

Matrix Operations Menu:
1. Input Matrix
2. Display Matrix
3. Add Two Matrices
4. Multiply Two Matrices
5. Transpose a Matrix
6. Exit
Enter your choice (1-6): 4
Product of Matrices:
19 22
43 50

Matrix Operations Menu:
1. Input Matrix
2. Display Matrix
3. Add Two Matrices
4. Multiply Two Matrices
5. Transpose a Matrix
6. Exit
Enter your choice (1-6): 6
Exiting program.
```

32. Define a class called Car with attributes such as make, Model & year. Include member functions to set & get these Attributes. Create an object of the Car class and Demonstrate the use of its member functions.

```cpp
#include <iostream>
#include <string>
using namespace std;
class Car {
private:
    string make;
    string model;
    int year;
public:
    // Member function to set the make
    void setMake(const string carMake) {
        make = carMake;
    }
    // Member function to get the make
    string getMake() const {
        return make;
    }
    void setModel(const string carModel) {
        model = carModel;
    }
    string getModel() const {
        return model;
    }
    void setYear(int carYear) {
        year = carYear;
    }
    int getYear() const {
        return year;
    }
    // Function to display the car details
    void display() const {
        cout << "Make: " << make << endl;
        cout << "Model: " << model << endl;
        cout << "Year: " << year << endl;
    }
};
int main() {
    Car myCar;
    // Set the car attributes
    myCar.setMake("Mahindra");
    myCar.setModel("XUV700");
    myCar.setYear(2025);
    // Display the car details
```

```
    myCar.display();
    // Get and display individual attributes
    cout << "Car Make: " << myCar.getMake() << endl;
    cout << "Car Model: " << myCar.getModel() << endl;
    cout << "Car Year: " << myCar.getYear() << endl;
return 0;
}
```

Output :-
Make: Mahindra
Model: XUV700
Year: 2025
Car Make: Mahindra
Car Model: XUV700
Car Year: 2025

33. Define a class 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.h>
using namespace std;
class Address{
    private :
        string street, city;
        int zipCode;
    public :
        void inputData(string str, string cty, int code){
            street = str;
            city = cty;
            zipCode = code;
        }
        void showData(){
            cout << "Street : " << street << endl;
            cout << "City : " << city << endl;
            cout << "zipCode : " << zipCode << endl;
        }
};
class Person{
    private :
        string name;
        Address;
```

```
    public :
        void inputData(string n, Address adr){
            name = n;
            address = adr;
        }
        void showData(){
            cout << "Name : " << name << endl;
            address.showData();
        }
};
int main(){
    Address myaddress;
    myaddress.inputData("123 Main Street", "Indore", 12345);

    Person myperson;
    myperson.inputData("John", myaddress);
    myperson.showData();
return 0;
}
```

Output :-
Name : John
Street : 123 Main Street
City : Indore
zipCode : 12345