

1. Write a C++ program to swap two numbers without using third variable.

```
#include <iostream.h>
#include <conio.h>

int main()
{
int a=5, b=10;
cout<<"Before swap a= "<<a<<" b= "<<b<<endl;
a=a*b; //a=50 (5*10)
b=a/b; //b=5 (50/10)
a=a/b; //a=10 (50/5)
cout<<"After swap a= "<<a<<" b= "<<b<<endl;
return 0;
}
```

Output:

Before swap a= 5 b= 10
After swap a= 10 b= 5

2. Write a C++ program to find all roots of a Quadratic Equation.

```
#include<iostream.h>
#include<math.h>
#include<conio.h>

int main() {
    int a = 1, b = 2, c = 1;
    float discriminant, realPart, imaginaryPart, x1, x2;
    if (a == 0) {
        cout << "This is not a quadratic equation";
    }else {
        discriminant = b*b - 4*a*c;
        if (discriminant > 0) {
            x1 = (-b + sqrt(discriminant)) / (2*a);
            x2 = (-b - sqrt(discriminant)) / (2*a);
            cout << "Roots are real and different." << endl;
            cout << "Root 1 = " << x1 << endl;
            cout << "Root 2 = " << x2 << endl;
        } else if (discriminant == 0) {
            cout << "Roots are real and same." << endl;
            x1 = (-b + sqrt(discriminant)) / (2*a);
            cout << "Root 1 = Root 2 =" << x1 << endl;
        }else {
            realPart = (float) -b/(2*a);
            imaginaryPart = sqrt(-discriminant)/(2*a);
            cout << "Roots are complex and different." << endl;
            cout << "Root 1 = " << realPart << " + " << imaginaryPart << "i"
            << endl;
            cout << "Root 2 = " << realPart << " - " << imaginaryPart << "i"
            << endl;
        }
    }
    return 0;
}
```

Output

Roots are real and same.

Root 1 = Root 2 = -1

3. Write a C++ program to print Fibonacci series.

```
#include <iostream.h>
#include<conio.h>

int main() {
    int n1=0,n2=1,n3,i,number;
    cout<<"Enter the number of elements: ";
    cin>>number;
    cout<<n1<<" "<<n2<<" "; //printing 0 and 1
    for(i=2;i<number;++i) //loop starts from 2 because 0 and 1 are already printed
    {
        n3=n1+n2;
        cout<<n3<<" ";
        n1=n2;
        n2=n3;
    }
    return 0;
}
```

Output:

Enter the number of elements: 10
0 1 1 2 3 5 8 13 21 34

4. Write a C++ program to check whether a number is prime or not.

```
#include <iostream.h>
#include<conio.h>

int main()
{
    int n, i, m=0, flag=0;
    cout << "Enter the Number to check Prime: ";
    cin >> n;
    m=n/2;
    for(i = 2; i <= m; i++)
    {
        if(n % i == 0)
        {
            cout<<"Number is not Prime."<<endl;
            flag=1;
            break;
        }
    }
    if(flag==0)
        cout << "Number is Prime."<<endl;
    return 0;
}
```

Output:

Enter the Number to check Prime: 17

Number is Prime.

Enter the Number to check Prime: 57

Number is not Prime.

5. Write a C++ program to add two Matrix.

```
#include<iostream.h>
#include<conio.h>
int main ()
{
    int m, n, p, q, i, j, A[5][5], B[5][5], C[5][5];
    cout<<"Enter rows and column of matrix A : ";
    cin>> m >> n;
    cout<<"Enter rows and column of matrix B : ";
    cin>> p >> q;
    if((m != p)&&(n != q))
    {
        cout<<"Matrices cannot be added!";
        exit(0);
    }
    cout<<"Enter elements of matrix A : ";
    for(i=0;i< m;i++)
        for(j =0; j < n;j++)
            cin>> A[i][j];
    cout<<"Enter elements of matrix B : ";
    for(i=0;i< p;i++)
        for(j =0; j < q;j++)
            cin>> B[i][j];
    for(i=0;i< m;i++)
        for(j =0; j < n;j++)
            C[i][j]= A[i][j]+ B[i][j];
    cout<<"Sum of matrices\n";
    for(i=0;i< m;i++)
    {
        for(j =0; j < n;j++)
            cout<< C[i][j]<<" ";
        cout<<"\n";
    }
    return0;
}
```

Output

Enter rows and column of matrix A : 2 2

Enter rows and column of matrix B : 2 2

Enter elements of matrix A : 1 2 3 4

Enter elements of matrix B : 4 3 2 1

Sum of matrices

5

5

Case 2 :

Enter rows and column of matrix A : 2 3

Enter rows and column of matrix B : 3 2

Matrices cannot be added!

Case 3 :

Enter rows and column of matrix A : 1 3

Enter rows and column of matrix B : 1 3

Enter elements of matrix A : 0 2 7

Enter elements of matrix B : 0 3 3

Sum of matrices

0 5 10

6. Write a C++ program to show Constructor and Destructor Example.

```
#include <iostream.h>
#include<conio.h>
class class_name
{
private:
    int a,b;
public:
    class_name(int aa, int bb)
    {
        cout<<"Constructor is called"<<endl;
        a = aa;
        b = bb;
        cout<<"Value of a: "<<a<<endl;
        cout<<"Value of b: "<<b<<endl;
        cout<<endl;
    }
    ~class_name()
    {
        cout<<"Destructor is called"<<endl;
        cout<<"Value of a: "<<a<<endl;
        cout<<"Value of b: "<<b<<endl;
    }
};

int main()
{
    class_nameobj(5,6);
    return 0;
}
```

Output
Constructor is called

Value of a: 5

Value of b: 6

Destructor is called

Value of a: 5

Value of b: 6

7. Write a C++ program to Display Student details using class.

```
#include <iostream.h>
#include <string.h>
#include <conio.h>

class Student
{
    private: std::string name;
    std::string studentClass;
    int rollNumber;
    double marks;

    public:
        Student(const std::string &studentName,
                const std::string &sClass, int rollNum, double studentMarks):
            name(studentName),
            studentClass(sClass),
            rollNumber(rollNum),
            marks(studentMarks) {}

    std::string calculateGrade() {
        if (marks >= 90) {
            return "A+";
        } else if (marks >= 80) {
            return "A";
        } else if (marks >= 70) {
            return "B";
        } else if (marks >= 60) {
            return "C";
        } else {
            return "D";
        }
    }
}
```

```
// Member function to display student information
void displayInformation() {
    std::cout << "\n\nName: " << name << std::endl;
    std::cout << "Class: " << studentClass << std::endl;
    std::cout << "Roll Number: " << rollNumber << std::endl;
    std::cout << "Marks: " << marks << std::endl;
    std::cout << "Grade: " << calculateGrade() << std::endl;
}

};

int main() {
    // Create a student object
    std::string studentName;
    std::string sClass;
    int rollNum;
    double studentMarks;
    std::cout << "Student details: ";
    std::cout << "\nName: ";
    std::getline(std::cin, studentName);
    std::cout << "Class: ";
    std::getline(std::cin, sClass);
    std::cout << "Roll number: ";
    std::cin >> rollNum;
    std::cout << "Marks (0-100): ";
    std::cin >> studentMarks;
    Student student(studentName, sClass, rollNum, studentMarks);
    student.displayInformation();
    return 0;
}
```

Output:

Student details:

Name: Ljubinka Marquita

Class: V

Roll number: 2

Marks (0-100): 98

Name: Ljubinka Marquita

Class: V

Roll Number: 2

Marks: 98

Grade: A+

Student details:

Name: Lia Chimezie

Class: VI

Roll number: 8

Marks (0-100): 88

Name: Lia Chimezie

Class: VI

Roll Number: 8

Marks: 88

Grade: A

8. Write a C++ program to demonstrate single inheritance.

```
#include <iostream.h>
#include <conio.h>
class base
{
public:
    int x;
    void getdata()
    {
        cout << "Enter the value of x = "; cin >> x;
    }
};

class derive : public base
{
private:
    int y;
public:
    void readdata()
    {
        cout << "Enter the value of y = "; cin >> y;
    }
    void product()
    {
        cout << "Product = " << x * y;
    }
};

int main()
{
    derive a;
    a.getdata();
    a.readdata();
```

```
a.product();  
    return 0;  
}
```

Output

```
Enter the value of x = 3  
Enter the value of y = 4  
Product = 12
```

9. Write a C++ program to find area class and over load area using function overloading.

```
#include<iostream.h>
#include<stdlib.h>
#include <conio.h>

float area(float r)
{
    return(3.14 * r * r);
}
float area(float b,float h)
{
    return(0.5 * b * h);
}
float area(float l,float b)
{
    return (l * b);
}
int main()
{
    float b,h,r,l;
    int ch;

    do
    {
        cout<<"\n\n *****Menu***** \n";
        cout<<"\n 1. Area of Circle";
        cout<<"\n 2. Area of Triangle";
        cout<<"\n 3. Area of Rectangle";
        cout<<"\n 4. Exit";
        cout<<"\n\n Enter Your Choice : ";
        cin>>ch;
```

```

switch(ch)
{
    case 1:
    {
        cout<<"\n Enter the Radius of Circle : ";
        cin>>r;
        cout<<"\n Area of Circle : "<<area(r);
        break;
    }
    case 2:
    {
        cout<<"\n Enter the Base & Height of Triangle :
";
        cin>>b>>h;
        cout<<"\n Area of Triangle : "<<area(b,h);
        break;
    }
    case 3:
    {
        cout<<"\n Enter the Length & Bredth of Rectangle
";
        cin>>l>>b;
        cout<<"\n Area of Rectangle : "<<area(l,b);
        break;
    }
    case 4:
        exit(0);
    default:
        cout<<"\n Invalid Choice... ";
    }
}while(ch!=4);
return 0;
}

```

Output:

1. Area of Circle
1. Area of Circle
2. Area of Triangle

3. Area of Rectangle

4. Exit

Enter Your choice: 1

Enter the Radius of Circle: 6

Area of circle: 113.04

2. Area of Triangle

1. Area of Circle

2. Area of Triangle

3. Area of Rectangle

4. Exit

Enter Your choice: 2

Enter the Base & Height of triangle: 6 5

Area of Triangle: 15

3. Area of Rectangle

1. Area of Circle

2. Area of Triangle

3. Area of Rectangle

4. Exit

Enter Your choice: 3

Enter the Length & Breadth of Triangle: 7 9

Area of Rectangle: 63

10. Write a C++ program to give example of constructors.

```
#include <iostream.h>
#include <conio.h>

class Employee
{
public:
    Employee()
    {
        cout<<"Default Constructor Invoked"<<endl;
    }
};

int main(void)
{
    Employee e1; //creating an object of Employee
    Employee e2;
    return 0;
}

Output:
Default Constructor Invoked
Default Constructor Invoked
```

11. Write a C++ program to show the use of friend function.

```
#include <iostream>
#include <conio.h>
```

```
class B;
class A
{
    int x;
public:
    void setdata(int i)
    {
        x=i;
    }
    friend void min(A,B);
};

class B
{
    int y;
public:
    void setdata(int i)
    {
        y=i;
    }
    friend void min(A,B);
};

void min(A a,B b)
{
    if(a.x<=b.y)
        std::cout << a.x << std::endl;
    else
        std::cout << b.y << std::endl;
}

int main()
```

```
{  
    A a;  
    B b;  
    a.setdata(10);  
    b.setdata(20);  
    min(a,b);  
    return 0;  
}
```

Output:

10

12. Write a C++ program to show working of constructor in inherited class.

```
#include <iostream.h>
classParent {
    intx;
public:
    Parent(inti)
    {
        x = i;
        cout << "Inside base class's parameterized "
            "constructor"
        << endl;
    }
};

classChild : publicParent {
public:
    Child(intx): Parent(x)
    {
        cout << "Inside sub class's parameterized "
            "constructor"
        << endl;
    }
};
intmain()
{
    return0;
}
Output:
Inside base class's parameterized constructor
Inside sub class's parameterized constructor
```

13. Write a C++ program to overload unary + operator.

```
#include <iostream.h>
#include <conio.h>

class Complex {
private:
    float real;
    float imag;

public:
    // Constructor to initialize real and imag to 0
    Complex() : real(0), imag(0) {}

    void input() {
        cout << "Enter real and imaginary parts respectively: ";
        cin >> real;
        cin >> imag;
    }

    // Overload the + operator
    Complex operator + (const Complex& obj) {
        Complex temp;
        temp.real = real + obj.real;
        temp.imag = imag + obj.imag;
        return temp;
    }

    void output() {
        if (imag < 0)
            cout << "Output Complex number: " << real << imag << "i";
        else
            cout << "Output Complex number: " << real << "+"
            << imag << "i";
    }
}
```

```
};

int main() {
    Complex complex1, complex2, result;

    cout << "Enter first complex number:\n";
    complex1.input();

    cout << "Enter second complex number:\n";
    complex2.input();

    // complex1 calls the operator function
    // complex2 is passed as an argument to the function
    result = complex1 + complex2;
    result.output();

    return 0;
}
```

Run Code

Output

Enter first complex number:

Enter real and imaginary parts respectively: 9 5

Enter second complex number:

Enter real and imaginary parts respectively: 7 6

Output Complex number: 16+11i

14. Write a C++ program to show how to achieve exception handling in C++.

```
#include <iostream>
#include <conio.h>

int main()
{
    double numerator, denominator, divide;
    cout << "Enter numerator: ";
    cin >> numerator;
    cout << "Enter denominator: ";
    cin >> denominator;
    try {
        if (denominator == 0)
            throw 0;

        divide = numerator / denominator; cout << numerator << " / " <<
        denominator << " = " << divide << endl;
    }

    catch (int num_exception) {
        cout << "Error: Cannot divide by " << num_exception << endl;
    }

    return 0;
}
```

Output 1

```
Enter numerator: 72
Enter denominator: 0
Error: Cannot divide by 0
```

Output 2

```
Enter numerator: 72
```

Enter denominator: 3

$72 / 3 = 24$

15. Write a C++ program to show function overloading in C++.

```
#include <iostream.h>
#include <conio.h>

class Cal {
public:
static int add(int a,int b){
    return a + b;
}
static int add(int a, int b, int c)
{
    return a + b + c;
}
};

int main(void) {
    Cal C;
    cout<<C.add(10, 20)<<endl;
    cout<<C.add(12, 20, 23);
    return 0;
}
```

Output:

30

55

16. Write a C++ program to demonstrate use of virtual function in C++.

```
#include <iostream.h>
class Base {
public:
    virtual void print() {
        cout << "Base Function" << endl;
    }
};

class Derived : public Base {
public:
    void print() {
        cout << "Derived Function" << endl;
    }
};

int main() {
    Derived derived1;

    Base* base1 = &derived1;

    base1->print();

    return 0;
}
```

Output
Derived Function

17. Write a C++ program to create a class Vehicle. Derived class are Two-Wheelers, Three-Wheelers. Display the properties of each type of vehicle using the member function of class.

```
#include<iostream.h>
#include<conio.h>
class vehicle
{
protected:
    char name[20];
    int wc;
public:
    void getdata()
    {
        cout<<"Enter Vehicle Name:- ";
        cin>>name;
        cout<<"Enter Wheel Count:- ";
        cin>>wc;
    }
    void outdata()
    {
        cout<<"\nVehicle is:- "<<name;
        cout<<"\nWheel Count:- "<<wc;
    }
};
class lightmotor:public vehicle
{
private:
    int sl;
public:
    void getdata()
    {
        vehicle::getdata();
    }
```

```

        cout<<"Enter Speed Limit:- ";
        cin>>sl;
    }
    void outdata()
    {
        vehicle::outdata();
        cout<<"\nSpeedLimit:- "<<sl;
    }
};

class heavymotor:public vehicle
{
private:
    int lc;
    char per[20];
public:
    void getdata()
    {
        cout<<endl<<endl;
        vehicle::getdata();
        cout<<"Enter Local Capacity:- ";
        cin>>lc;
        cout<<"Enter Permit:- ";
        cin>>per;
    }
    void outdata()
    {
        vehicle::outdata();
        cout<<"\nLocalCapacity:- "<<lc;
        cout<<"\nPermit:- "<<per;
    }
};
void main()
{
    lightmotor l;
    heavymotor h;
    clrscr();
    l.getdata();
}

```

```
l.outdata();
h.getdata();
h.outdata();
getch();
}
```

18. Write a C++ program to display student Marks sheet using Multiple Inheritance.

```
#include <iostream.h>
class Vehicle {
public:
    Vehicle() { cout << "This is a Vehicle\n"; }
};

class fourWheeler : public Vehicle {
public:
    fourWheeler()
    {
        cout << "Objects with 4 wheels are vehicles\n";
    }
};

class Car : public fourWheeler {
public:
    Car() { cout << "Car has 4 Wheels\n"; }
};

int main()
{
    Car obj;
    return 0;
}
```

Output

This is a Vehicle
Objects with 4 wheels are vehicles
Car has 4 Wheels

19. Write a C++ program to display student Marks sheet using Multiple Inheritance.

```
#include <iostream.h>
using namespace std;
class student
{
public:
    char a[30];
    int roll;
    void get()
    {
        cout<<"Enter the name:"<<endl;
        cin>>a;
        cout<<"Enter the roll.no:"<<endl;
        cin>>roll;
    }
};
class mark
{
public:
    int mark[4],i;
    void in()
    {
        cout<<"Enter the marks:"<<endl;
        for(i=0;i<4;i++)
        {
            cin>>mark[i];
        }
    }
};
class process:public student,public mark
{
public:
    int t;
```

```

float avg;
void calc()
{
    t=mark[0]+mark[1]+mark[2]+mark[3];
    avg=t/4;
}
void dis()
{
    cout<<"Name:"<<a<<endl;
    cout<<"Roll.no:"<<roll<<endl;
    cout<<"Marks entered:";
    for(i=0;i<4;i++)
    {
        cout<<mark[i]<<" ";
    }
    cout<<endl;
    cout<<"Total marks:"<<t<<endl;
    cout<<"Average:"<<avg<<endl;
}
int main()
{
    cout<<"\t\tStudent mark list using multiple inheritance"<<endl;
    cout<<"\t\t_____"
    endl;
    process v;
    v.get();
    v.in();
    v.calc();
    v.dis();
    return 0;
}

```

Output
Student mark list using multiple inheritance

Enter the name : Rameshkumar
Enter the roll.no : 87
Enter the marks : 90 89 85 95
Name :Rameshkumar
Roll.no : 87
Marks entered :90 89 85 95
Total marks :359
Average :89

20. Write a C++ program to print reverse of any given no using class.

Example input no=2345 output no=5432.

```
#include<iostream.h>
int main ()
{
//variables initialization
int num, reverse = 0, rem;
num=1234;
cout << "\nThe number is" << num;
while(num!=0)
{
    rem = num%10;
    reverse = reverse *10+ rem;
    num/=10;
}
cout << "Reversed Number: " << reverse;
getch();
return 0;
}
```

Output

Reversed Number : 4321

The number is: 1234

Reversed Number: 4321

21. Write a C++ program to illustrate the use of pure virtual function in polymorphism.

```
#include <iostream.h>
class Shape {
protected:
    float dimension;
public:
    void getDimension() {
        cin>> dimension;
    }
    virtual float calculateArea() = 0;
};
class Square : public Shape {
public:
    float calculateArea() {
        return dimension * dimension;
    }
};
class Circle : public Shape {
public:
    float calculateArea() {
        return 3.14 * dimension * dimension;
    }
};
int main() {
    Square square;
    Circle circle;
    cout << "Enter the length of the square: ";
    square.getDimension();
    cout << "Area of square: " <<square.calculateArea() <<endl;
    cout << "\nEnter radius of the circle: ";
    circle.getDimension();
    cout << "Area of circle: " <<circle.calculateArea() <<endl;
    return 0;
}
```

}

Output

Enter the length of the square: 4

Area of square: 16

Enter radius of the circle: 5

Area of circle: 78.5

22. Write a C++ program that finds factorial of a given number using loop.

```
#include<iostream.h>
#include<conio.h>
int main()
{
    int I,n,fact=1;
    clrscr();
    cout<<"Enter a no to find factorial=";
    cin>>n;
    for(i=1;i<=n;i++)
        fact=fact*I;
    cout<<"Factorial ="<<fact;
    getch();
}
```

Output.
Enter a no to find factorial=6

120

23. Write a C++ Program to Calculate the Power of a Number

```
#include <iostream.h>
#include <math.h>
int main()
{
    float base, exp, power;
    cout << "Enter the base: ";
    cin >> base;
    cout << "Enter the exponent: ";
    cin >> exp;
    power = pow(base, exp);
    cout << base << "^" << exp << " = " << power << endl;
    return 0;
}
```

Output

```
Enter the base: 2
Enter the exponent: 5
2^5 = 32
```

24. Write a C++ program to find Cube Root of input number.

```
#include <iostream.h>
#include <math.h>

int main(){
    float number, ans;
    clrscr();
    cout << "Enter any number: ";
    cin >> number;
    ans = cbrt(number);
    cout << "\n Cube Root of " << number << " is: " << ans;
    getch();
}
```

Output

Enter any number: 27
Cube Root of 27 is: 3

25. Write a C++ program to find input number is palindrome or not.

```
#include <iostream.h>
#include<conio.h>
int main()
{
    int n, num, digit, rev = 0;
    clrscr();
    cout << "Enter a positive number: ";
    cin>>num;
    n = num;
    do
    {
        digit = num % 10;
        rev = (rev * 10) + digit;
        num = num / 10;
    } while (num != 0);
    cout << " The reverse of the number is: " << rev << endl;
    if (n == rev)
        cout << " The number is a palindrome.";
    else
        cout << " The number is not a palindrome.";
    getch();
    return 0;
}
```

Output

Enter a positive number: 12321
The reverse of the number is: 12321
The number is a palindrome.

Enter a positive number: 12331
The reverse of the number is: 13321

The number is not a palindrome.

26.

Write a C++ program to count word in a sentence.

```
#include <iostream.h>
#include<conio.h>
int main()
{
    string sentence = "Mary had a little lamb";
    int words = 0;
    int lenOfSentence = sentence.size();
    clrscr();
    for(int i = 0; i<lenOfSentence; i++)
    {
        if(sentence[i] == ' ')
        {
            words++;
        }
    }
    words = words + 1;
    cout << "No. of words = " << words << endl;
}
```

Output

No. of words = 5

27. Write a C++ program to that finds largest of three number.

```
#include <bits/stdc++.h>
#include <bits/stdc++.h>

intmain()
{
    inta, b, c;
    cout << "Enter the three numbers a, b & c" << endl;
    cin >> a >> b >> c;
    if(a >= b)
    {
        if(a >= c)
    {
        cout << "The Largest Among Three Numbers is : "
            << a << endl;
    }
    else{
        cout << "The Largest Among Three Numbers is : "
            << c << endl;
    }
}
else{
    if(b >= c) {
        cout << "The Largest Among Three Numbers is : "
            << b << endl;
    }
    else{
        cout << "The Largest Among Three Numbers is : " << c << endl;
    }
}
return0;
}
```

Output
Enter the three numbers a, b & c
The Largest Among Three Numbers is : 4196384

27. Write a C++ program to check input number is Armstrong or not.

```
#include<iostream.h>
#include<conio.h>
int main ()
{
    int num, temp, rem, sum = 0;
    clrscr();
    cout << "Enter number to be checked : ";
    cin >> num;
    temp = num;
    while (temp != 0)
    {
        rem = temp % 10;
        sum = sum + rem * rem * rem;
        temp = temp / 10;
    }
    if (sum == num)
        cout << "\n" << num << " is an Armstrong number.";
    else
        cout << "\n" << num << " is not an Armstrong number.";
    getch();
    return 0;
}
```

Output
Enter number to be checked : 371
371 is an Armstrong number.

28. Write a C++ program to print alphabet triangle.

```
#include <iostream.h>
#include <conio.h>

int main()
{
    char ch='A';
    int i, j, k, m;
    for(i=1;i<=5;i++)
    {
        for(j=5;j>=i;j--)
            cout<<" ";
        for(k=1;k<=i;k++)
            cout<<ch++;
        ch--;
        for(m=1;m<i;m++)
            cout<<--ch;
        cout<<"\n";
        ch='A';
    }
    return 0;
}
```

Output:

```
A
ABA
ABCBA
ABCDCBA
ABCDEDCBA
```

29. Write a C++ program to convert a string of characters into upper or lower case.

```
#include <iostream.h>
#include <conio.h>

void lower_string(string str)
{
    for(int i=0;str[i]!='\0';i++)
    {
        if (str[i] >= 'A' && str[i] <= 'Z') //checking for uppercase
characters
            str[i] = str[i] + 32;      //converting uppercase to
lowercase
    }
    cout<<"\n The string in lower case: "<< str;
}

void upper_string(string str)
{
    for(int i=0;str[i]!='\0';i++)
    {
        if (str[i] >= 'a' && str[i] <= 'z') //checking for lowercase
characters
            str[i] = str[i] - 32;      //converting lowercase to
uppercase
    }
    cout<<"\n The string in upper case: "<< str;
}
```

```
int main()
{
    string str;
    cout<<"Enter the string ";
    getline(cin,str);
    lower_string(str); //function call to convert to lowercase
    upper_string(str); //function call to convert to uppercase
    return 0;
}
```

Output:

```
Enter the string Hola Amigos!
The string in lower case: hola amigos!
The string in upper case: HOLAMIGOS!
```

30. Write a c++ program to read a string and find the number of vowels in it

```
#include<iostream>
#include<conio.h>
```

```
intmain()
{
    charstr[100] = "prepinsta";
    int vowels = 0;
    clrscr();
    for(int i=0; str[i]; i++)
    {
        if(str[i] == 'a' || str[i] == 'e' || str[i] == 'i' || str[i] == 'o' || str[i] == 'u'
           || str[i] == 'A' || str[i] == 'E' || str[i] == 'I' || str[i] == 'O' || str[i] == 'U')
        {
            vowels++;
        }
    }
}
```

```
cout << "Total Vowels : " << vowels;
```

```
return 0;
}
```

Output

Total Vowels : 3