Lab-File

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"
<<end;
      cout << "Root 2 = " <<realPart<< " - " <<imaginaryPart<< "i"
<<end;
    }
  }
  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 alread
y 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
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,
conststd::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<<"\nVehicleis:- "<<name;
                cout<<"\nWheelCount:- "<<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>
classVehicle {
public:
   Vehicle() { cout << "This is a Vehicle\n"; }
};

classfourWheeler : publicVehicle {
public:
   fourWheeler()
   {
      cout << "Objects with 4 wheels are vehicles\n";
   }
};
classCar : publicfourWheeler {
public:
   Car() { cout << "Car has 4 Wheels\n"; }
};

intmain()
{
   Car obj;
   return0;
}
```

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:publicstudent,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
intnum, 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();
return0;
}
```

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;
clracr();
  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;
cltrscr();
    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: HOLA AMIGOS!
```

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(inti=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;

return0;
}
```

Output

Total Vowels : 3

Topic Date

Ques WAP a program to print fibonacci series?

```
# Include < stdio.h >
# Include < conio.h >
main ()
{
int a=0, b=1, c, n;

Printf ("Enter the number");
Scanf ("%d", &n);

Printf ("%d %d", a, b);
for ( i=0; i<n; i++)
{
C=a+b;
a=b;
b=c;
}
Printf ("%d", c);

getch ();
}
```

Output → Enter the number → 2

012

Topic _____ Date _____

Ques To read n numbers and display it.

```
# include <stdio.h>
int main ()
{
int n, i;
Printf ("Enter the number of element:");
Scanf ("%d", &n);

int numbers [n];
Printf ("Enter %d numbers: \n", n);
for (i=0; i<n; i++)
 {
 Scanf ("%d", &numbers [i]);
 }

Printf ("Entered numbers are:");
for (i = 0; i<n; i++)
{
Printf ("%d", numbers [i]);
 }

return 0;
}
```

Output

```
Enter the number of elements : 3
 34, 56, 90
 Entered numbers are : 34, 56, 90
```

Topic _____ Date _____

Output :-

Sum of array elements = 15.

Topic _____ Date _____

Ques To demonstrate the concept of one dimensional array finding the sum of array elements

```
# include < stdio.h >
int main ()
{
int arr [ ] = { 1, 2, 3, 4, 5 };
int sum = 0;
for ( int i = 0; i < sizeof (arr) / sizeof (arr[0]);
    i++);
{
  sum += arr[i];
  }
Printf (" Sum of array element : %d \n ", sum);

return 0;
 }
```

Topic _____ Date _____

Output

Updated array : 1, 2, 3, 4, 5.

Topic _____ Date _____

Ques Write a program to search an element in array

```
# include < stdio.h >
main()
{
int i, n, search, a[10];
Printf (" Enter the number to be searched");
Scanf (" %d", & search);

for( i = 0; i<10; i++)
Scanf (" %d", &a[i]);

for (i=0; i<10; i++)
{
  if (a[i] == search)
{
Printf ("%d=[%d]", search, i+1);
break;
}}

if (i==n)
{
Printf ("%d", search);
}

return 0;
}
```

Topic _____ Date _____

Output :-

~~The~~ Enter the no of Matrix A $\begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}$

Enter the no of Matrix B $\begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}$

$\begin{bmatrix} 4 & 4 \\ 4 & 4 \end{bmatrix}$

Write a program to sum of two matrix :-

```
#Include <stdio.h>
main()
{
int a[2][2], b[2][2], c[2][2], i, j;
Printf("Enter the first Matrix");
for(i=0; i<2; i++)
{
for(j=0; j<2; j++)
{
Printf Scanf("%d", &a[i][j]);
}}
Printf("Enter the second Matrix");
for(i=0; i<2; i++)
{
for(j=0; j<2; j++)
{
Scanf("%d", &b[i][j]);
}
for(i=0; i<2; i++)
{
for(j=0; j<2; j++)
{
c[i][j] = a[i][j] + b[i][j]
}}.
```

Soma
29-09-23

Topic _____ Date _____

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

  printf ("%d", c[i][j]);
}}

return 0;
}
```

Output :

Enter the number to be searched

3 [3]

1

2

3

2

8

Topic

Date

Ques Write a program of multiply of two matrix

```
# include <stdio.h>
# include <conio.h>
main()
{
int a[2][2] ,b[2][2],c[2][2], i, j, k;
printf (" Enter the no of Matrix A ");
for (i=0; i<2; i++)
{
for( j=0; j<2; j++)
{
scanf ("%d" ,& a[i][j]);
}
}
printf (" Enter the no. of Matrix B");
for (i=0; i<2; i++)
{
for( j=0; j<2; j++)
{
scanf ("%d", & b[i][j]);
}}
for( i=0; i<2; i++)
{
for( j=0; j<2; j++)
{
```

```
c[i][j]=0;
for(k=0; k<2; k++)
{
c[i][j]= a[i][k] * b[k][j];
}}
for(i=0; i<2; i++)
{
for(j=0; j<2; j++)
{
printf("%d", c[i][j]);
}
printf("\n");
}
getch();
}
```

Output of transpose

Enter the matrix $\begin{bmatrix} 2 & 3 \\ 3 & 2 \end{bmatrix}$

Transpose of matrix $\begin{bmatrix} 2 & 3 \\ 3 & 2 \end{bmatrix}$

Ques:

Topic

Date

Ques Write a program to concatenate two string

```
# include < stdio.h>
# include < string.h>

int main()
{
char str1[50] = "Hello";
char str2[] = "World!";

strcat (str1, str2);

printf ("Concatenated string : %s\n", str1);

return 0;
}
```

Output

Concatenated string : Hello World!

Topic _____ Date _____

Output

Source : Hello, World!

Destination : Hello, World!

Topic

Date

Ques To copy a string into another string in C

```
#include <stdio.h>
#include <string.h>
int main()
{
char source[] = "Hello, World!";
char destination[20];
strcpy (destination, source);
Printf ("Source : %.s\n" source);
Printf ("Destination : %.s\n", destination);
return 0;
}
```

Topic

Date

Output -1.

11 -> 7 -> 3 -> NULL.

Topic _____ Date _____

Ques Implementation of linklist using an array.

```
# include < stdio . h >
# define Max - size 100
struct node {
int data;
int inden;
};

struct node [ Max_ size ];
int nent pointers [ Max _ size ];
int head = -1;
int freelist = 0;
void initalized linkedlist ( ) {
for ( int i = 0; i < Max- size -1; i++ ) {
 nent pointers [ i ] = i+1;
 }
// nent pointers [Max_ size -1] = -1;
}
void insert At Begining (int data ) {
if ( free list == -1) {
Printf (" Linked list is full, cannot insert
        more elements . \n ");
return ;
}
int new node inden = free list;
free list = nent pointers [ free list ];
nodes [ new node inden ] . data = data;
```

```
nodes[newnodeindex].next = head;
head = newnodeindex;
}

void displayLinkedList(){
int current = head;
while(current != -1){
printf("%d -> ", nodes[current].data);
current = nodes[current].next;
}
printf("NULL\n");
}

int main(){
initializedLinkedList();
insertAtBegining(3);
insertAtBegining(7);
insertAtBegining(11);

displayLinkedList();

return 0;
}
```

Ques. Implementation of stack using array in C.

```
# include < stdio.h>
# include < stdlib.h>
# define Max_size 100
struct stack {
int arr [ Max_size ];
int top;
};
void initalize ( struct stack * stack ) {
    stack -> top = -1;
}
int isEmpty ( struct stack * stack ) {
return ( stack > top == Max_size -1);
}
void Push ( struct stack * stack, int value) {
if ( is full (stack)) {
Printf (" Stack overflow \n");
return;
}

stack > arr [ ++ stack > top ] = value;
}

int pop ( struct stack * stack)
{
```

```
if (isEmpty(stack)){
Printf("stack underflow\n");
exit(Exit_failure);
}
return stack->arr[stack->top--];
}
int peek(struct stack* stack){
if(isEmpty(stack)){
Printf("stack is empty\n");
exit(Exit_failure);
}
return stack->arr[stack->top];
}
int main(){
struct stack mystack;
initialize(&my stack);
Push(&my stack,10);
Push(&my stack,20);
Push(&my stack,30);

Printf("Top element:%d\n", peek(&my stack));
Printf("Popped element:%d\n", pop(&my stack));
Printf("Popped element:%d\n", pop(&my stack));
Printf("Is the stack empty?%s\n", is Empty
(&my stack)? "Yes": "No");

return 0;
}
```

Topic _____ Date _____

Topic _____ Date _____

Output =:

Top element : 30

Popped element : 30

Popped element : 20

Is the stack empty? No.

Topic _____ Date _____

Ques Calculate factorial of number using recursive fun.

```
#include <stdio.h>
int factorial (int n){
if (n==0 || n==1){

return 1;
}
else{
return n * factorial (n-1);
}}
int main (){
int num = 5;
Printf (" Factorial of %d is %d\n", num, factorial
                (num));
return 0;
}
```

Output :-

Factorial of 5 is 120.

Topic _____ Date _____

Que Implementation of queue using an array

```
# include < stdio.h>
# include < stdlib.h>
# define Max_size 100
struct Queue {
int front, rear, size;
unsigned capacity;
int * array;
};

struct Queue * create Queue ( Unsigned Capacity ){
struct Queue * queue = ( struct Queue* ) malloc
    ( size of ( struct Queue ));

queue> capacity = Capacity;
queue > front = queue > size = 0;
queue > rear = capacity - 1;
queue > array = ( int*) malloc ( queue> Capacity *
        size of ( int ));
return queue;
}
int is full ( struct Queue * queue){
return ( queue > size == queue > capacity );
}
```

```
int is empty (struct Queue *queue){
return (queue -> size == 0);
}
void enqueue (struct Queue *queue, int item){
if (is full (queue))
return;
queue -> rear = (queue -> rear + 1) % queue -> capacity
queue -> array [queue -> rear] = item;
queue -> size ++;
}
int dequeue (struct Queue *queue){
if (is empty (queue))
return -1;
int item = queue -> array [queue -> front];
queue -> front = (queue -> front + 1) % queue -> capacity;
queue -> size --;
return item;
}
int front (struct Queue* queue){
if (is empty (queue))
return -1;
return queue -> array [queue -> front];
}
int rear (struct Queue *queue){
if (is empty (queue))
return -1;
return queue -> array [queue -> rear];
}
```

```
{
int main(){
struct Queue *queue = CreateQueue(Max_Size);

enqueue(queue, 10);
enqueue(queue, 20);
enqueue(queue, 30);

Printf("Front element is: %d\n", front(queue));
Printf("Rear element is: %d\n", rear(queue));

Printf("Dequeue item is: %d\n", dequeue(queue));

Printf("Front element is: %d\n", front(queue));
Printf("Rear element is: %d\n", rear(queue));

return 0;
}
```

Output :-

```
Front element : 10
Rear element : 30
Dequeued item is : 10
Front element is : 20
Rear element is : 30
```

Topic

Date

Ques Implementation of binary search tree using array in C.

```
#include <stdio.h>
#include <stdlib.h>
#define Max_size 100
struct node{
  int data;
  struct Node* left;
  struct Node* right;
};

struct Node* createNode(int data){
struct Node* newNode = (struct Node*) malloc(sizeof
                        (struct Node));
newnode->data = data;
newnode->left = newnode->right = NULL;
 return newNode;
}
struct Node* insert(struct Node* root, int data
{
  if (root == NULL)
return createNode(data);
if (data < root->data).
root->left = insert (root->left, data);
else if (data > root->data)
root->right = insert (root->right, d
```

```
return root;
}
void inorderTraversal(struct Node* root){
  if (root != NULL){
inorderTraversal(root->left);
printf("%d", root->data);
inorderTraversal(root->right);
  }}

int main(){
int arr[max_size];
int size = 0;

size = insert(arr, size, 30);
size = insert(arr, size, 20);
size = insert(arr, size, 40);
size = insert(arr, size, 10);
size = insert(arr, size, 25);

printf("Inorder Traversal : ");
inorderTraversal(arr, size);
  return 0;
  }
```

Topic _____ Date _____

Ans. Arrange the list of numbers in ascending order using Bubble sort.

```
# include < stdio.h>
void bubble sort (int arr[ ], int n) {

for (int i=0; i<n-1; i++)
 for (int j=0; j<n-i-1; j++) {
  if (arr [j] > arr [j+1]) {
  int temp = arr [j];
   arr [j] = arr [j+1];
   arr[ j+1]= temp;
    }
  }
}

int main ()
{
int arr [ ]= {64, 25, 12, 22, 11};
int n= size of (arr) / size of ( arr[0]);
 bubble sort ( arr, n);
Printf (" sorted array: ");
 for (int i =0; i<n; i++)
Printf ("%d", arr [i]);
 return 0;
 }
```

Output → Sorted array : 11 12 22 25 64.

Topic _____ Date _____

Q1. 11. Write a program to sort a list of numbers in ascending

Topic _____ Date _____

Ques. Arrange the list of numbers in ascending order using insertion sort.

```
#include <stdio.h>
void insertionSort(int arr[], int n){
  int i, key, j;
  for(i=1; i<n; i++){
    key = arr[i];
    j = i-1;

  while (j>=0 && arr[j] > key){
    arr[j+1] = arr[j];
    j = j-1;
  }
    arr[j+1] = key;
void printArray(int arr[], int size){
for(int i=0; i<size; i++)
  {
    printf("%d", arr[i]);
  }
  printf("\n");
  }
  int main(){
  int arr[] = {12, 11, 13, 5, 6};
  int n = sizeof(ar)/sizeof(arr[0]);
  printf("Original array:");
  printArray(Arr, n);
```

Topic _____ Date _____

```
insertionSort (arr, n);
printf (" Sorted array: ");
printArray (arr, n);
return 0;
}
```

Output :-

Original array : 12 11 13 5 & 6.

Sorted array : 5 6 11 12 13.

Topic _____ Date _____

Ques Arrange the list of numbers in ascending order using selection sort in C.

```
# include < stdio.h>
void selection sort( int arr[ ], int n){
 int i, j, minIndex, temp;
 for( i=0; i<n-1; i++) {
  minIndex = i;
  for( j = i+1; j < n; j++)
  {
    if (arr[j]< arr[ minIndex ]){
   minIndex = j;
  } }
  temp = arr[ i ];
  arr[ i ]= arr[ minIndex ];
  arr [ minIndex ]= temp;
 }}
 int main( ){
 int arr[ ]= { 64, 25, 12, 22, 11 };
 int n = size of (arr) / size of( arr [0]);
  Printf (" original array:");
  for (int i=0; i < n; i++){
  Printf (" %d", arr [ i ]);
  }
```

Topic _____ Date _____

selectionSort (arr, n);

printf ("\n Sorted array in ascending order");

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

printf ("%d", arr[i]);

}

return 0;

}

Output :-

Original array : 64, 25, 12, 22, 11

Sorted array in ascending order : 11, 12, 22, 25, 64.

Topic _____ Date _____

Ques. Arrange the list of numbers in ascending order using Merge Sort.

```
# include <stdio.h>
void merge(int arr[], int l, int m, int r){
int i, j, k;
int n1 = m - l + 1;
int n2 = r - m;

int L[n1], R[n2];
for(i=0; i<n1; i++)
   L[i] = arr[l + i];
for(j=0; j<n2; j++)
   R[j] = arr[m + 1 + j];

i=0;
j=0;
k=l;
while(i<n1 && j<n2){
if (L[i] <= R[j]){
arr[k] = L[i];
  i++;
}
  else{
  arr[k] = R[j];
     j++;
}
   k++;
```

```
}

while (i<n1){
arr[k]=L[i];
i++;
k++;
}
while (j<n2){
arr[k]=R[j];
j++;
k++;
}}

void merge sort (int arr[], int l, int r){
if (l<r){
int m=l+(r-l)/2;
merge sort(arr, l, m);
merge sort(arr, m+1, r);
merge (arr, l, m, r);
}}

int main()
{
int arr[]={12, 11, 13, 5, 6, 7};
int arr_size = size of (arr) / size of(arr[0]
printf("Given array is \n");
for(int i=0; i<arr_size; i++)
```

```
printf("%d", arr[i]);
printf("\n");
mergesort(arr, 0, arr_size - 1);
printf("sorted array is\n");
for(int i=0; i<arr_size); i++
printf("%d", arr[i]);
printf("\n");
return 0;
}
```

Output :-

```
Given array is
12, 11, 13, 5, 6, 7

Sorted array is
5, 6, 7, 11, 12, 13.
```

Ques Arrange the list of numbers in ascending order using Quick sort.

```
# include < stdio.h>
void swap (int *a, int *b){
 int t = *a;
*a = *b;
 *b = t;
}
int partition (int arr[], int low, int high){
int pivot = arr [high];
int i = (low -1);
for (int j= low; j <= high -1; j++){
 if (arr[j] < pivot ){
 i++;
swap (&arr[i], &arr[j]);
 }
}
swap (&arr [i+1], &arr [high]);
 return (i+1);
}
void quick sort (int arr[], int low, int high)
{
 if ( low < high ) {
int pi = partition (arr, low, high);
quick sort( arr, low, pi-1);
quick sort( arr, pi+1, high);
} }
```

```
int main()
{
int arr[ ]={64, 25, 12, 22, 11};
int n = size of (arr) / size of (arr[0]);
printf ("%d", *arr[i]);
}

quick sort (arr, 0, n-1);
printf ("\n sorted array : ");
for (int i=0; i<n; i++){
printf ("%d", *arr[i]);
}
return 0;
}
```

Output :-

Original array : 64, 25, 12, 22, 11
Sorted array : 11, 12, 22, 25, 64.