SUBMITTED BY:	Prachi Aggarwal
COURSE:	Bsc(H) Computer Science
ROLL NO:	21570015
SUBJECT:	Discrete Mathematics
UNIQUE PAPER CODE:	32341202
SEMESTER:	IInd
YEAR:	Ist
SUBMITTED TO:	Reena Jain
TOPIC:	Practical File

Q1) Write a Program to create a SET A and determine the cardinality of SET for an input array of elements (repetition allowed) and perform the following operations on the SET:

a) **ismember (a, A): check whether an element belongs to set or not and return value as true/false.**
b) **powerset(A): list all the elements of power set of A.**

Ans)

```
#include<iostream>
#include<cmath>
using namespace std;
char arr[20];
void menu(char a[],char a1[],int i,int i2);
int input(char a[]);
bool is_member(int n1,char ch, char a[]);
void subset(char arr[],int i);
void binary(int n,int num,char arr[]);
int main()
{
        int size1;
  int choice;
  char ch;
  cout<<"Enter your choice : 1)is_member 2)powerset"<<endl;
  cin>>choice;
  if(choice==1)
  {
    size1= input(arr);
    cout<<"Enter the element to which you want to check"<<endl;
    cin>>ch;
    bool f;
```

```
        f=is_member(size1,ch,arr);
        if(f==1)
        {
            cout<<"The character is in the set"<<endl;
        }
        else{
            cout<<"The character is not in the set"<<endl;
        }
    }
    else if(choice==2)
    {
        size1=input(arr);
        subset(arr,size1);
    }
            return 0;
}
int input(char a[])
{
            int i;
    cout<<"Enter the elements of your set and terminates by entering character '%'"<<endl;
    cin>>a[0];
    for(i=1; a[i-1]!='%';i++){
        cin>>a[i];
    }
    return i-1;
}
bool is_member(int n1,char ch, char a[])
{
```

```
        bool flag=false;
    for(int i=0; i<n1; i++)
    {
        if(ch==a[i])
        {
            flag=true;
        }
    }
    if(flag==true){
        return true;
    }
    else{
        return false;
    }
}
void binary(int n,int num,char arr[])
{
    int a1[num];
    int a2[num];
    while(n!=0)
    {
        for(int i=0; i<num; i++)
        {
            if(n%2==0)
            {
                a1[i]=0;
                n=n/2;
            }
            else
```

```
            {
                a1[i]=1;
                n=n/2;
            }
        }
    }
    int a=0;
    for(int i=num-1; i>=0; i--)
    {
        a2[a]=a1[i];
        if(a2[a]==1)
        {
            cout<<arr[a]<<" ";
        }
        a++;
    }
    return;
}
void subset(char arr[], int i)
{
    int subNo = pow(2,i);
    for(int j=0; j<subNo ; j++)
    {
        cout<<"{";
        binary(j,i,arr);
        cout<<"}";
        cout<<endl;
    }
    return ;
```

}

OUTPUT

```
Enter your choice : 1)is_member 2)powerset
1
Enter the elements of your set and terminates by entering character '%'
abcde%
Enter the element to which you want to check
a
The character is in the set

--------------------------------
Process exited after 6.62 seconds with return value 0
Press any key to continue . . .
```

```
Enter your choice : 1)is_member 2)powerset
2
Enter the elements of your set and terminates by entering character '%'
abc%
{}
{c }
{b }
{b c }
{a }
{a c }
{a b }
{a b c }

--------------------------------
Process exited after 3.408 seconds with return value 0
Press any key to continue . . .
```

```
Enter your choice : 1)is_member 2)powerset
3
Wrong choice

--------------------------------
Process exited after 1.565 seconds with return value 0
Press any key to continue . . .
```

Q2.Create a class SET and take two sets as input from user to perform following SET Operations:

a) **Subset: Check whether one set is a subset of other or not.**
b) **Union and Intersection of two Sets.**
c) **Complement: Assume Universal Set as per the input elements from the user.**
d) **Set Difference and Symmetric Difference between two SETS**
e) **Cartesian Product of Sets.**

Ans)

```
#include<iostream>

#include<cmath>

using namespace std;

char arr[20];

char arr1[20];

void menu(char a[],char a1[],int i,int i2);

int input(char a[]);

class SET

{

	private:

		int n1;

		int n2;

	public:

		SET(){

			n1=0;

			n2=0;

		}

		SET(int s1,int s2)

		{

			n1=s1;

			n2=s2;

		}
```

```
void is_subset(int n1,int n2 ,char a[],char a1[])
{

int i,j,k;
int count=0;
for(i=0; i<n1; i++)
        {
        for(j=0; j<n2; j++)
{
if(a[i]==a1[j])
{
        count++;
}
}
}
if(count==n1)
{
cout<<"set1 is the subset of set2"<<endl;
}
else
{
cout<<"set1 is the not the subset of set2"<<endl;
}
}
void Cartesian(int n1, int n2, char a[] , char a1[])
{
        cout<<"The cartesian product of set1 and set2 is : "<<endl;
        cout<<"{";
        for(int i=0; i<n1; i++)
```

```
        {
                for(int j=0; j<n2; j++)
                {
                        cout<<"("<<a[i]<<","<<a1[j]<<") , ";
                }
        }
        cout<<"}"<<endl;
}
void Intersection(int n1, int n2, char a[] , char a1[])
{
        int p=0;
int q,r;
char inter[n1+n2];
for(q=0; q<n2; q++)
{
for(r=0; r<n1; r++)
{
if(arr1[q]==arr[r])
{
   inter[p]=arr1[q];
   p++;
}
}
}
cout<<"\nThe intersection of the sets are:\n";
for(int aa=0; aa<p; aa++)
{
cout<<inter[aa]<<" ";
}
```

```
}
void Union(int n1,int n2,char a[], char a1[])
{

int i,j,k;
char un[n1+n2];
int count=0;
for(i=0; i<n1; i++){
un[i]=a[i];
count++;
}
for(j=0; j<n2; j++){
bool flag=false;
for(k=0; k<n1; k++){
if(a1[j]==a[k]){
        k=n1;
        flag=true;
}
}
if(flag==false)
{
un[i]=a1[j];
count++;
i++;
}
}
for(int m=0; m<count; m++)
{
cout<<un[m]<<" ";
```

```
}



}
void difference(int n1, int n2, char a[], char a1[])
{
int i=0;
int j,k;
char differ[n1+n2];
for(j=0; j<n1; j++)
{
bool flag=false;
for(k=0; k<n2; k++)
{
if(a[j]==a1[k])
{
        k=n2;
        flag=true;
}
}
if(flag == false)
{
differ[i]=a[j];
i++;
}
}
cout<<"The difference of set1-set2 sets are:\n";
for(int r=0; r<i; r++)
{
```

```
cout<<differ[r]<<" ";
}
}
void symmetric(int n1, int n2 , char a[], char a1[])
{
int i=0;
int j,k;
int aa,b;
int c=0;
char differ[n1+n2];
char differ1[n1+n2];
for(j=0; j<n1; j++)
{
bool flag=false;
for(k=0; k<n2; k++)
{
if(a[j]==a1[k])
{
        k=n2;
        flag=true;
}
}
if(flag == false)
{
differ[i]=a[j];
i++;
}
}
for(aa=0; aa<n2; aa++)
```

```
                {
                    bool flagi=false;
                for(b=0; b<n1; b++)
                {
                    if(a1[aa]==a[b])
                    {
                        b=n1;
                        flagi=true;
                }
                }
                if(flagi == false)
                {
                    differ1[c]=a1[aa];
                c++;
                }
                }
                Union(i,c,differ,differ1);
            }



};
int main()
{
            int size1,size2;
            size1 = input(arr);
            size2 = input(arr1);
            menu(arr,arr1,size1,size2);
            return 0;
}
```

```
int input(char a[])
{
        int i;
   cout<<"Enter the elements of your set and terminates by entering character '%'"<<endl;
   cin>>a[0];
   for(i=1; a[i-1]!='%';i++){
      cin>>a[i];
   }
   return i-1;
}
void menu(char a[],char a1[],int i,int i2)
{
   int choice;
   char choose;
   do{
      cout<<"Enter your choice\n1)Union & Intersection\n2)Difference & Symmetric
Difference\n3)Check_Subset or not\n4)Complement\n5)Cartesian Product\n";
      cin>>choice;
      if(choice==1){
         SET s;
         cout<<"The union of two sets are\n";
         s.Union(i,i2,a,a1);
                              s.Intersection(i,i2,a,a1);
      }
      else if(choice==2){
         SET s;
         s.difference(i,i2,a,a1);
         cout<<"\nsymmetric difference"<<endl;
         s.symmetric(i,i2,a,a1);
      }
```

```
        else if(choice==3)
        {
            SET s;
            s.is_subset(i,i2,arr,arr1);
        }

        else if(choice==4)
        {
            cout<<"the array two is universal set"<<endl;
            cout<<"COMPLIMENT"<<endl;
            SET s;
            s.difference(i,i2,a,a1);

        }
                    else if(choice==5)
                    {
                            SET s;
                            s.Cartesian(i,i2,a,a1);
                    }
        else
        {
            cout<<"INVALID CHOICE   enter correct choice"<<endl;
        }

        cout<<"\nDo you want to choose more?(y/n):"<<endl;
        cin>>choose;
    }while(choose=='y');
    return;
}
```

OUTPUT

```
D:\prachi_c\Q2.exe
Enter the elements of your set and terminates by entering character '%'
abcdef%
Enter the elements of your set and terminates by entering character '%'
abc%
Enter your choice
1)Union & Intersection
2)Difference & Symmetric Difference
3)Check_Subset or not
4)Complement
5)Cartesian Product
1
The union of two sets are
a b c d e f
The intersection of the sets are:
a b c
Do you want to choose more?(y/n):
y
Enter your choice
1)Union & Intersection
2)Difference & Symmetric Difference
3)Check_Subset or not
4)Complement
5)Cartesian Product
2
The difference of set1-set2 sets are:
d e f
symmetric difference
d e f
Do you want to choose more?(y/n):
y
Enter your choice
1)Union & Intersection
2)Difference & Symmetric Difference
3)Check_Subset or not
4)Complement
5)Cartesian Product
3
set1 is the not the subset of set2

Do you want to choose more?(y/n):
y
Enter your choice
1)Union & Intersection
2)Difference & Symmetric Difference
3)Check_Subset or not
4)Complement
5)Cartesian Product
4
```

```
4
the array two is universal set
COMPLIMENT
The difference of set1-set2 sets are:
d e f
Do you want to choose more?(y/n):
y
Enter your choice
1)Union & Intersection
2)Difference & Symmetric Difference
3)Check_Subset or not
4)Complement
5)Cartesian Product
5
The cartesian product of set1 and set2 is :
{(a,a) , (a,b) , (a,c) , (b,a) , (b,b) , (b,c) , (c,a) , (c,b) , (c,c) , (d,a) , (d,b) , (d,c) , (e,a) , (e,b) , (e,c) , (f,a) , (f,b) , (f,c) , }

Do you want to choose more?(y/n):
y
Enter your choice
1)Union & Intersection
2)Difference & Symmetric Difference
3)Check_Subset or not
4)Complement
5)Cartesian Product
6
INVALID CHOICE   enter correct choice

Do you want to choose more?(y/n):
n

--------------------------------
Process exited after 45.14 seconds with return value 0
Press any key to continue . . .
```

Q3. Create a class RELATION, use Matrix notation to represent a relation. Include functions to check if a relation is reflexive, Symmetric, Anti-symmetric and Transitive. Write a Program to use this class.

Q4.Use the functions defined in Ques 3 to find check whether the given relation is:

a) **Equivalent, or**
b) **Partial Order relation, or**
c) **None**

Combined Ans3+4)

```
#include<iostream>

using namespace std;

bool f,f1,f2,f3;

class RELATION

{

public:

   int size;

   char A[][20];



   RELATION()

   {

      size = 0;

   }

   void matric()

   {

      int p;

      char arr[20];

      cout << "Enter the elements of set and terminates by #" << endl;

      cin >> arr[0];
```

```
for (p = 1; arr[p - 1] != '#'; p++)
{
    cin >> arr[p];
}
size = p - 1;
cout << "The cardinality of the set is: " << size << endl;
for(int i=0; i<size; i++){
    for(int j=0; j<size ; j++)
    {
        A[i][j]='0';
    }
}
bool flag = true;
cout << "Enter the relation A->A:" << endl;
char ch;
char R[1][2];
int index1 =0, index2 =0;
while (flag == true)
{
    int count = 0;
    char a, b;
    cin >> R[0][0];
    cin >> R[0][1];
    a = R[0][0];
    b = R[0][1];
    int index1 =0, index2 =0;
    for (int i = 0; i < size; i++)

    {
```

```
        if (a == arr[i])
        {
            index1 = i;
        }
        if (b == arr[i])
        {
            index2 = i;
        }

    }
    A[index1][index2]='1';

    cout << "Do you want to choose more (y/n)" << endl;
    cin >> ch;
    if (ch == 'n')
    {
        flag = false;
    }
}

for (int i = 0; i < size; i++)
{
    for (int j = 0; j < size; j++)
    {
        cout << A[i][j] <<" ";
    }
    cout << endl;
}
```

```
}
void reflexive()
{
    f = true;
    for (int i = 0; i < size; i++)
    {
        if (A[i][i] == '0')
        {
            f = false;


        }
    }
    if (f == false)
    {
        cout << "The given relation is not reflexive:" << endl;
    }
    else
    {
        cout << "The given relation is REFLEXIVE:" << endl;
    }
}
void symmetric(){
    f1=true;
    for(int i=0; i<size; i++)
    {
        for(int j=0 ; j<size ; j++)
        {
            if(A[i][j]!=A[j][i]){
```

```
            f1=false;
        }
    }
  }
  if (f1 == false)
  {
    cout << "The given relation is not symmetric" << endl;
  }
  else
  {
    cout << "The given relation is SYMMETRIC:" << endl;
  }
}
void antisymmetric(){
  f2=true;
  for(int i=0; i<size; i++)
  {
    for(int j=0 ; j<size ; j++)
    {
      if(A[i][j]=='1'&&A[j][i]=='1'&&i!=j){
        f2=false;
      }
    }
  }
  if (f2 == false)
  {
    cout << "The given relation is not anti symmetric" << endl;
  }
  else
```

```
    {
        cout << "The given relation is ANTI SYMMETRIC:" << endl;
    }
}
void transitive(){
        f3=true;
        for(int i=0; i<size; i++){
                for(int j=0; j<size ; j++)
                {
                        for(int k=0; k<size; k++)
                        {
                                if(A[i][j]=='1'&&A[j][k]=='1'&&A[i][k]!='1'){

                                        f3= false;
                                        break;
                                }
                                }
                        }
                }
                if (f3 == false)
    {
        cout << "The given relation is not transitive" << endl;
    }
    else
    {
        cout << "The given relation is TRANSITIVE" << endl;
    }
        }
        bool equivalent(){
```

```
        if(f&&f1&&f3){
                cout<<"Equivalent relation"<<endl;
                return 1;
                }
                else{
                        cout<<"not equivalent relation"<<endl;
                        return 0;
                }
        }
        bool porel(){
        if(f&&f2&&f3){
                cout<<"Partial ordered relation"<<endl;
                return 1;
                }
                else{
                        cout<<"not partial ordered relation"<<endl;
                        return 0;
                }
        }
};
int main()
{
   RELATION r;
   r.matric();
   r.reflexive();
   r.symmetric();
   r.antisymmetric();
   r.transitive();
   bool e = r.equivalent();
```

```
    bool p = r.porel();
    if(e!=1&&p!=1){
        cout<<"the relation is neither equivalent nor porel"<<endl;
        }
    return 0;
}
```

OUTPUT

```
D:\prachi_c\Q3_Q4.exe
Enter the elements of set and terminates by #
abcd#
The cardinality of the set is: 4
Enter the relation A->A:
aa
Do you want to choose more (y/n)
y bb
Do you want to choose more (y/n)
y cc
Do you want to choose more (y/n)
y dd
Do you want to choose more (y/n)
y ac
Do you want to choose more (y/n)
y ad
Do you want to choose more (y/n)
y cd
Do you want to choose more (y/n)
y ca
Do you want to choose more (y/n)
y da
Do you want to choose more (y/n)
y dc
Do you want to choose more (y/n)
n
1 0 1 1
0 1 0 0
1 0 1 1
1 0 1 1
The given relation is REFLEXIVE:
The given relation is SYMMETRIC:
The given relation is not anti symmetric
The given relation is TRANSITIVE
Equivalent relation
not partial ordered relation

--------------------------------
Process exited after 147.5 seconds with return value 3221225477
Press any key to continue . . .
```

Output 2

```
D:\prachi_c\Q3_Q4.exe
y ee
Do you want to choose more (y/n)
y ab
Do you want to choose more (y/n)
y ca
Do you want to choose more (y/n)
y cd
Do you want to choose more (y/n)
y cb
Do you want to choose more (y/n)
y ce
Do you want to choose more (y/n)
y ed
Do you want to choose more (y/n)
n
1 1 0 0 0
0 1 0 0 0
1 1 1 1 1
0 0 0 1 0
0 0 0 1 1
The given relation is REFLEXIVE:
The given relation is not symmetric
The given relation is ANTI SYMMETRIC:
The given relation is TRANSITIVE
not equivalent relation
Partial ordered relation

--------------------------------
Process exited after 43.78 seconds with return value
Press any key to continue . . .
```

Q5) Write a Program to generate the Fibonacci Series using recursion.

Ans)

```
#include<iostream>
using namespace std;
int fib(int a)
{
   if((a==1)||(a==0))
   {
      return a;
   }
   else{
      return (fib(a-1)+fib(a-2));
   }
}
int main()
{
   int num , i;
   cout<<"Enter the limit of the series"<<endl;
   cin>>num;
   cout<<"FIBONACCI SERIES\n";
   for(i=0; i<num; i++)
   {
      cout<<" "<<fib(i);
   }
   return 0;
}
```

Output

```
Enter the limit of the series
8
FIBONACCI SERIES
 0 1 1 2 3 5 8 13
--------------------------------
Process exited after 0.9848 seconds with return value 0
Press any key to continue . . .
```

Q6) Write a Program to implement Tower of Hanoi using recursion.

Ans)

```
#include<iostream>
using namespace std;
void toh(int n, char a1, char a2, char a3)
{
   if(n==0)
   {
      return ;
   }
   else{
      toh(n-1,a1,a3,a2);
      cout<<"The disk "<<n<<" move from rod "<<a1<<" to rod "<<a2<<endl;
      toh(n-1,a3,a2,a1);
   }
}
int main()
{
         cout<<"********TOWER OF HANOI********"<<endl;
   int disk;
   cout<<"Enter number of disks"<<endl;
   cin>>disk;
   toh(disk,'A','B','C');
   return 0;
}
```

Output

```
D:\prachi_c\Q6.exe
*******TOWER OF HANOI*******
Enter number of disks
3
The disk 1 move from rod A to rod B
The disk 2 move from rod A to rod C
The disk 1 move from rod B to rod C
The disk 3 move from rod A to rod B
The disk 1 move from rod C to rod A
The disk 2 move from rod C to rod B
The disk 1 move from rod A to rod B

--------------------------------
Process exited after 1.406 seconds with return value 0
Press any key to continue . . .
```

Q7) Write a Program to implement binary search using recursion.

Ans)

```
#include<iostream>
using namespace std;
int BinarySearch(int arr[], int n , int i , int end)
{
   int middle;
   middle = (i+end)/2;
   if(i>end)
   {
      cout<<"The number is not present in the list"<<endl;
      exit(0);
   }
   else{

      if(n==arr[middle])
      {
         return middle;
      }
      else if(arr[middle]>n){
         return BinarySearch(arr,n,i,middle-1);
      }
      else
      {
         return BinarySearch(arr,n,middle+1,end);
      }
   }
}
int main()
```

```
{
    cout<<"Enter the size of array:"<<endl;
    int size;
    cin>>size;
    int arr[size];
    cout<<"Enter the content of array but in sorted manner:"<<endl;
    for(int i=0; i<size ;i++)
    {
        cin>>arr[i];
    }
    cout<<"your array is"<<endl;
    for(int i=0;i<size; i++)
    {
        cout<<arr[i]<<" ";
    }
    cout<<"\nStart Searching"<<endl;
    int front=0;
    int back=size-1;
    int num;
    cout<<"Enter the number want to search"<<endl;
    cin>>num;
    int index;
    index=BinarySearch(arr,num,front,back);
    cout<<"The element is found at index: "<<index<<endl;
    return 0;
}
```

Output

```
D:\prachi_c\Q7.exe
Enter the size of array:
4
Enter the content of array but in sorted manner:
1 2 3 4
your array is
1 2 3 4
Start Searching
Enter the number want to search
4
The element is found at index: 3

--------------------------------
Process exited after 15.58 seconds with return value 0
Press any key to continue . . .
```

Q8) Write a Program to implement Bubble Sort. Find the number of comparisons during each pass and display the intermediate result.

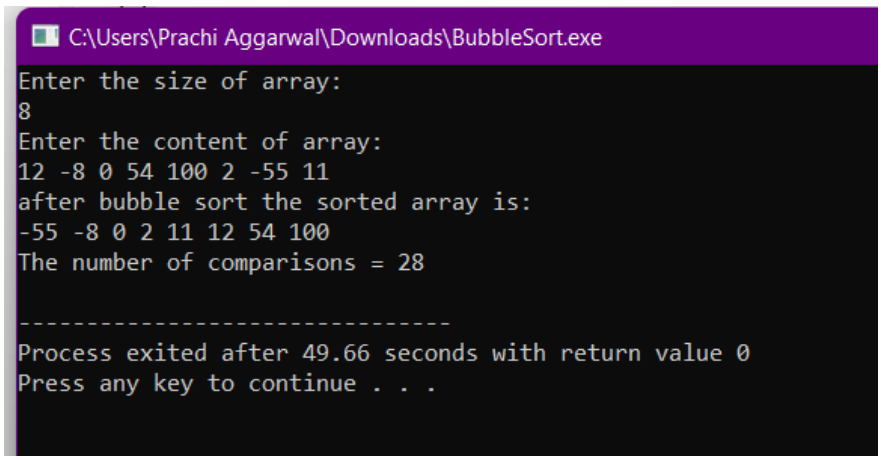
Ans)

```
#include<iostream>
using namespace std;
int main()
{
   cout<<"Enter the size of array:"<<endl;
   int size;
   int counter=0;
   cin>>size;
   int arr[size];
   cout<<"Enter the content of array:"<<endl;
   for(int i=0; i<size ;i++)
   {
      cin>>arr[i];
   }
   for(int i=0; i<size; i++)
   {
      for(int j=0; j<size-i-1; j++)
      {
          counter++;
         if(arr[j]>arr[j+1])
         {
            int temp = arr[j];
            arr[j] = arr[j+1];
            arr[j+1] = temp;
         }
      }
   }
```

```
    cout<<"after bubble sort the sorted array is:"<<endl;
    for(int i=0;i<size; i++)
    {
        cout<<arr[i]<<" ";
    }
    cout<<"\nThe number of comparisons = "<<counter<<endl;

}
```

Output



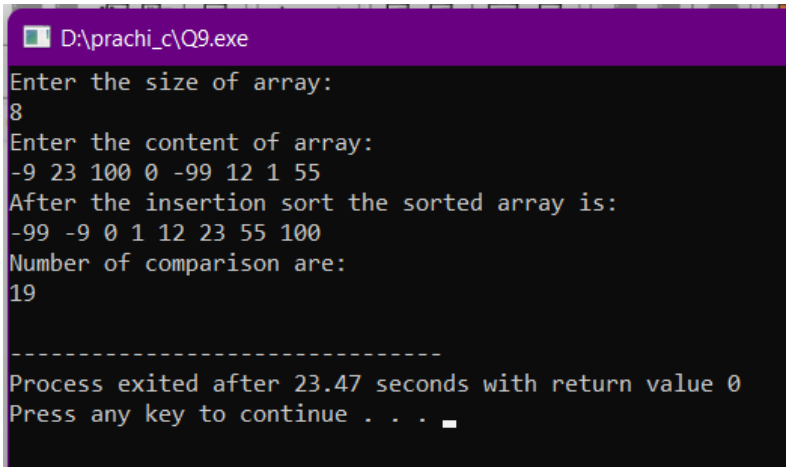
Q9) Write a Program to implement Insertion Sort. Find the number of comparisons during each pass and display the intermediate result. Use the observed values to plot a graph to analyse the complexity of algorithm.

Ans)

```
#include<iostream>
using namespace std;
int main()
{
   cout<<"Enter the size of array:"<<endl;
   int size;
   cin>>size;
   int arr[size];
   int comp=0;
   cout<<"Enter the content of array:"<<endl;
   for(int i=0; i<size ;i++)
   {
      cin>>arr[i];
   }
   for(int i =1 ;i<size ; i++)
   {
      int current = arr[i];
      int j = i-1;
      comp++;
      while( j >= 0 && current < arr[j])
      {
         arr[j+1] = arr[j];
         j--;
         comp++;
      }
      arr[j+1]=current;
```

```
    }
    cout<<"After the insertion sort the sorted array is:"<<endl;
    for(int i=0 ; i<size ; i++){
        cout<<arr[i]<<" ";
    }
    cout<<"\nNumber of comparison are:"<<endl;
    cout<<comp<<endl;
}
```

Output



Q10) Write a Program that generates all the permutations of a given set of digits, with or without repetition. (For example, if the given set is {1,2}, the permutations are 12 and 21). (One method is given in Liu)

Ans)

```
#include<iostream>
using namespace std;
void permut(string a,int l ,int r)
{
   if(l==r){
      cout<<a<<endl;
   }
   else{
      for(int i=l ; i<=r; i++)
      {
         swap(a[l],a[i]);
         permut(a,l+1,r);
         swap(a[l],a[i]);
      }
   }
}
int main()
{
   string str;
   cout<<"Enter your string want to permut"<<endl;
   cin>>str;
   permut(str,0,str.length()-1);
   return 0;
}
```

Output

```
D:\prachi_c\Q10.exe
Enter your string want to permut
abc
The possible permutations are
abc
acb
bac
bca
cba
cab
```

Q11) Write a Program to calculate Permutation and Combination for an input value n and r using recursive formula of ^nC_r and ^nP_r

Ans)

```
#include<iostream>
using namespace std;
int fact(int n);
int main()
{
   int n;
   cout<<"Enter the total number of elements:"<<endl;
   cin>>n;
   int r;
   cout<<"Enter number of elements arranged or select:"<<endl;
   cin>>r;
   cout<<"Permutation is : nPr"<<endl;
   int P;
   P = fact(n)/fact(n-r);
   cout<<P<<endl;
   int C;
   cout << "Combination is : nCr"<<endl;
   C = fact(n)/(fact(r)*fact(n-r));
   cout<<C<<endl;
}
int fact(int n)
{
   if(n==0)
   {
      return 1;
   }
```

```
    else{
        return n*fact(n-1);
    }
}
```

Output

```
D:\prachi_c\Q11.exe
Enter the total number of elements:
5
Enter number of elements arranged or select:
3
Permutation is : nPr
60
Combination is : nCr
10

--------------------------------
Process exited after 38.95 seconds with return value 0
Press any key to continue . . .
```

Q12) For any number n, write a program to list all the solutions of the equation $x_1 + x_2 + x_3 + \ldots + x_n = C$, where C is a constant (C<=10) and $x_1, x_2, x_3, \ldots, x_n$ are nonnegative integers using brute force strategy

Ans)

```
#include <iostream>

using namespace std;

class Combos

{

        public:

  // Function to diaplay combination

                void display(int b[], int n)

                {

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

        cout << b[i] << " ";

        }


  // Function to calculte possible combination

                int combos(int b[], int k, int n, int s)

                {

                        if (k == 0){

                                b[k] = s;

                                display(b, n);

                                cout << "\n";

                                return 0;

        }


        for (int i = 0; i <= s; i++)

                        {

                                b[k] = i;
```

```
                                        combos(b, k - 1, n, s - i);
            }
            }
};
int main()
{
    Combos obj;
    int s, n;
    cout << "Enter the no. of groups ::"<<endl;
    cin >> n;
    cout << "Enter the sum ::";
    cin >> s;
    int b[n];
    obj.combos(b, n - 1, n, s);
    return 0;
}
```

Output

D:\prachi_c\Q12.exe

```
Enter the no. of groups ::
3
Enter the sum ::5
5 0 0
4 1 0
3 2 0
2 3 0
1 4 0
0 5 0
4 0 1
3 1 1
2 2 1
1 3 1
0 4 1
3 0 2
2 1 2
1 2 2
0 3 2
2 0 3
1 1 3
0 2 3
1 0 4
0 1 4
0 0 5

--------------------------------
Process exited after 94.52 seconds with return value 0
Press any key to continue . . .
```

Q13) Write a Program to accept the truth values of variables x and y, and print the truth table of the following logical operations:

a) **Conjunction**
b) **Disjunction**
c) **Exclusive OR**
d) **Conditional**
e) **Bi-conditional**
f) **Exclusive NOR**
g) **Negation**
h) **NAND**
i) **NOR**

Ans)

```
#include <iostream>
using namespace std;
int main()
{
    int x, y;
    char ch;
    int choice;
    cout << "Enter the value of x and y" << endl;
    cin >> x >> y;
    do
    {
        cout << "Enter the choice\n1)Conjunction\n2)Disjunction\n3)Exclusive OR\n4)Conditioinal\n5)Bi-Conditional\n6)Exclusive NOR\n7)Negation\n8)NAND\n9)NOR\n"
          << endl;
        cin >> choice;
        switch (choice)
        {
        case 1:
            cout << "Conjuction of x and y is: " << (x & y) << endl;
            break;
        case 2:
            cout << "Disjuction of x and y is: " << (x | y) << endl;
```

```
        break;
    case 3:
        cout << "Exclusive OR of x and y is: " << (x ^ y) << endl;
        break;
    case 4:
        if (x == 1 && y == 0)
        {
            cout << "Coniditional of x and y is: " << 0 << endl;
        }
        else
        {
            cout << "Coniditional of x and y is: " << 1 << endl;
        }
        break;
    case 5:
        if ((x == 1 && y == 1) || (x == 0 && y == 0))
        {
            cout << "BiConiditional of x and y is: " << 1 << endl;
        }
        else
        {
            cout << "BiConiditional of x and y is: " << 0 << endl;
        }
        break;
    case 6:
        cout << "Exclusive NOR of x and y is: " << !(x ^ y) << endl;
        break;
    case 7: cout<<"negation of x is: "<<!x<<endl;
        cout<<"negation of y is: "<<!y<<endl;
```

```
            break;
        case 8: if(x==1&&y==1)
            {
                cout<<"Nand of x and y is: "<<0<<endl;
            }else{
                cout<<"Nand of x and y is: "<<1<<endl;
            }
            break;
        case 9: if(x==0&&y==0)
            {
                cout<<"Nor of x and y is: "<<1<<endl;
            }else{
                cout<<"Nor of x and y is: "<<0<<endl;
            }
        }
        cout<<"Do you want to choose more: (y/n)"<<endl;
        cin>>ch;
    }while(ch=='y');
    return 0;
}
```

Output

D:\prachi_c\Q13.exe

```
Enter the value of x and y
1 0
Enter the choice
1)Conjunction
2)Disjunction
3)Exclusive OR
4)Conditioinal
5)Bi-Conditional
6)Exclusive NOR
7)Negation
8)NAND
9)NOR

1
Conjuction of x and y is: 0
Do you want to choose more: (y/n)
y
Enter the choice
1)Conjunction
2)Disjunction
3)Exclusive OR
4)Conditioinal
5)Bi-Conditional
6)Exclusive NOR
7)Negation
8)NAND
9)NOR

2
Disjuction of x and y is: 1
Do you want to choose more: (y/n)
y
Enter the choice
1)Conjunction
2)Disjunction
3)Exclusive OR
4)Conditioinal
5)Bi-Conditional
6)Exclusive NOR
7)Negation
8)NAND
9)NOR

3
Exclusive OR of x and y is: 1
Do you want to choose more: (y/n)
y
Enter the choice
```

```
y
Enter the choice
1)Conjunction
2)Disjunction
3)Exclusive OR
4)Conditioinal
5)Bi-Conditional
6)Exclusive NOR
7)Negation
8)NAND
9)NOR

4
Coniditional of x and y is: 0
Do you want to choose more: (y/n)
y
Enter the choice
1)Conjunction
2)Disjunction
3)Exclusive OR
4)Conditioinal
5)Bi-Conditional
6)Exclusive NOR
7)Negation
8)NAND
9)NOR

5
BiConiditional of x and y is: 0
Do you want to choose more: (y/n)
y
Enter the choice
1)Conjunction
2)Disjunction
3)Exclusive OR
4)Conditioinal
5)Bi-Conditional
6)Exclusive NOR
7)Negation
8)NAND
9)NOR

6
Exclusive NOR of x and y is: 0
Do you want to choose more: (y/n)
y
Enter the choice
```

```
9)NOR

6
Exclusive NOR of x and y is: 0
Do you want to choose more: (y/n)
y
Enter the choice
1)Conjunction
2)Disjunction
3)Exclusive OR
4)Conditioinal
5)Bi-Conditional
6)Exclusive NOR
7)Negation
8)NAND
9)NOR

7
negation of x is: 0
negation of y is: 1
Do you want to choose more: (y/n)
y
Enter the choice
1)Conjunction
2)Disjunction
3)Exclusive OR
4)Conditioinal
5)Bi-Conditional
6)Exclusive NOR
7)Negation
8)NAND
9)NOR

8
Nand of x and y is: 1
Do you want to choose more: (y/n)
y
Enter the choice
1)Conjunction
2)Disjunction
3)Exclusive OR
4)Conditioinal
5)Bi-Conditional
6)Exclusive NOR
7)Negation
8)NAND
9)NOR

9
```

```
9
Nor of x and y is: 0
Do you want to choose more: (y/n)
n

--------------------------------
Process exited after 59.35 seconds with return value 0
Press any key to continue . . .
```

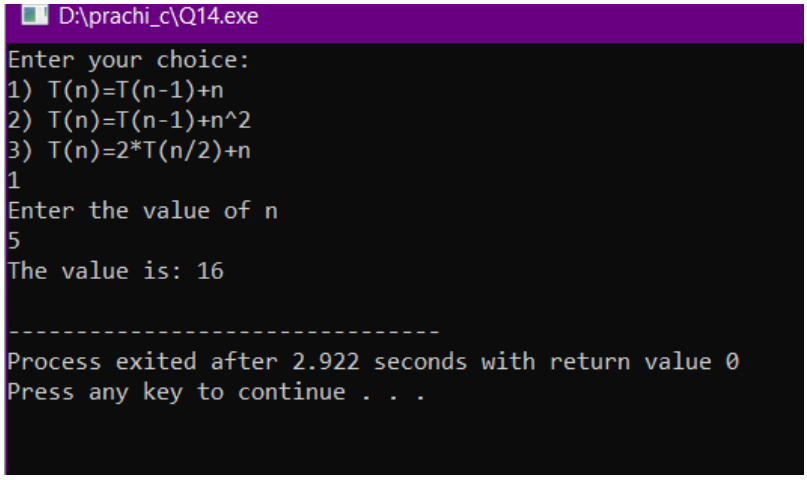
Q14) Write a program to accept an input `n` from the user and graphically represent the values of `T(n)` where `n` varies from `0` to `n` for the recurrence relations. For e.g. T(n) = T(n-1) + n, T(0) = 1, T(n) = T(n-1) + n^2, T(0) =1, T(n) = 2*T(n)/2 + n, T(1)=1.

Ans)

```
#include<iostream>
#include<cmath>
using namespace std;
int v1;
int T1(int n);
int T2(int n);
int T3(int n);
int main()
{
   int n;
   int ch;
   cout<<"Enter your choice:\n1) T(n)=T(n-1)+n\n2) T(n)=T(n-1)+n^2\n3) T(n)=2*T(n/2)+n"<<endl;
   cin>>ch;
   cout<<"Enter the value of n"<<endl;
   cin>>n;
   int value;
   switch(ch)
   {
      case 1: value=T1(n);
              cout<<"The value is: "<<value<<endl;
              break;
      case 2: value=T2(n);
              cout<<"The value is: "<<value<<endl;
              break;
      case 3: value=T3(n);
              cout<<"The value is: "<<value<<endl;
              break;
   }
}
int T1(int n)
{
   if(n==0){
      return 1;
   }
   else{
      return T1(n-1)+n;
   }
```

```
}
int T2(int n)
{
   if(n==0)
   {
      return 1;
   }
   else{
      return T2(n-1)+pow(n,2);
   }
}
int T3(int n)
{
   if(n==1)
   {
      return 1;
   }
   else{
      return 2*T3(n/2)+n;
   }
}
```

Output



```
D:\prachi_c\Q14.exe
Enter your choice:
1) T(n)=T(n-1)+n
2) T(n)=T(n-1)+n^2
3) T(n)=2*T(n/2)+n
2
Enter the value of n
5
The value is: 56

--------------------------------
Process exited after 1.952 seconds with return value 0
Press any key to continue . . .
```

```
D:\prachi_c\Q14.exe
Enter your choice:
1) T(n)=T(n-1)+n
2) T(n)=T(n-1)+n^2
3) T(n)=2*T(n/2)+n
3
Enter the value of n
5
The value is: 13

--------------------------------
Process exited after 3.374 seconds with return value 0
Press any key to continue . . .
```

Q15) Write a Program to store a function (polynomial/exponential), and then evaluate the polynomial. (For example store `f(x) = 4n`^{3} `+ 2n + 9` in an array and for a given value of `n`, say `n = 5`, evaluate (i.e. compute the value of `f(5)`).

Ans)

```
#include<iostream>
#include<cmath>
using namespace std;
int polynomial(int n , int d, int arr[]);
int main()
{
   int degree;
   int n;
   int result;
   cout<<"Enter the degree of the polynomial"<<endl;
   cin>>degree;
   int arr[degree+1];
   cout<<"Enter the coefficients of the equation"<<endl;
   for(int i=0; i<=degree; i++)
   {
      cin>>arr[i];
   }
   cout<<"Enter the value of n in f(n) at which want to find the solution"<<endl;
   cin>>n;
   result=polynomial(n,degree,arr);
   cout<<"The solution of the equation is"<<endl;
   int d=degree;
   for(int i=0; i<=degree; i++)
   {
      cout<<arr[i]<<"n^"<<d<<" + ";
```

```
        d--;
    }
    cout<<" with the value of n= "<<n<<" is "<<endl;
    cout<<result<<endl;
    return 0;
}
int polynomial(int n, int d , int arr[])
{
    int equal;
    int sum=0;
    equal=d;
    for(int i=0; i<=d; i++)
    {
        sum+= arr[i]*pow(n,equal);
        --equal;
    }
    return sum;
}
```

Output

```
D:\prachi_c\Q15.exe
Enter the degree of the polynomial
3
Enter the coefficients of the equation
2 -3 1 5
Enter the value of n in f(n) at which want to find the solution
1
The solution of the equation is
2n^3 + -3n^2 + 1n^1 + 5n^0 +  with the value of n= 1 is
5

--------------------------------
Process exited after 19.5 seconds with return value 0
Press any key to continue . . .
```

Q16) Write a Program to represent Graphs using the Adjacency Matrices and check if it is a complete graph.

Ans)

```
#include<iostream>
using namespace std;
void adjancecy(char V[]);
void completeGraph(char A[][10],int n);
int main()
{
   char v[10];
   adjancecy(v);
   return 0;
}
void adjancecy(char V[])
{
   int num ;
   char A[10][10];
   cout<<"Enter the number of vertices "<<endl;
   cin>>num;
   char a;
   cout<<"Enter the name of the vertices"<<endl;
   for(int i=0; i<num ;i++)
   {
      cin>>V[i];
   }
   for(int a=0; a<num; a++)
   {
      for(int b=0; b<num ; b++)
      {
```

```
        A[a][b]='0';
    }
}
cout<<"Enter the pair of vertices for edges one by one"<<endl;
char ch;
char R[1][2];
int index1 , index2;
bool flag=true;
while(flag==true)
{
    int count=0;
    char a,b;
    cin>>R[0][0];
    cin>>R[0][1];
    a=R[0][0];
    b=R[0][1];
    bool ff=false;
    for(int i=0; i<num ;i++)
    {
        for(int j=0; j<num; j++)
        {
            if(a==V[i]&&b==V[j])
            {
                ff=true;
                break;
            }
        }
    }
    if(ff==false)
```

```
    {
        cout<<"Wrong pair of vertices enter again"<<endl;
    }
    index1=0 ;
    index2=0;
    for(int i=0; i<num; i++)
    {
        if(a==V[i])
        {
            index1=i;
        }
        if(b==V[i]){
            index2=i;
        }
    }
    A[index1][index2]='1';
    A[index2][index1]='1';
    cout<<"Do you want to choose more(y/n)"<<endl;
    cin>>ch;
    if(ch=='n')
    {
        flag=false;
    }
}
cout<<"The adjacency matrix is : "<<endl;
for(int i=0; i<num ;i++)
{
    for(int j=0; j<num; j++)
    {
```

```
            cout<<A[i][j]<<" ";
        }
        cout<<endl;
    }
    completeGraph(A,num);
}
void completeGraph(char A[][10],int n)
{
    bool f=true;
    for(int i=0; i<n; i++)
    {
        for(int j=0; j<n; j++)
        {
            if(A[i][j]!='1'&&(i!=j))
            {
                f=false;
                break;
            }
        }
    }
    if(f==false)
    {
        cout<<"The given graph is not the complete graph"<<endl;
    }
    else {
        cout<<"The given graph is complete graph"<<endl;
    }
}
```

Output

```
D:\prachi_c\Q16.exe
Enter the number of vertices
5
Enter the name of the vertices
abcde
Enter the pair of vertices for edges one by one
aa
Do you want to choose more(y/n)
y ab
Do you want to choose more(y/n)
y ac
Do you want to choose more(y/n)
y ad
Do you want to choose more(y/n)
y ae
Do you want to choose more(y/n)
y bc
Do you want to choose more(y/n)
y bd
Do you want to choose more(y/n)
y be
Do you want to choose more(y/n)
y cd
Do you want to choose more(y/n)
y ce
Do you want to choose more(y/n)
y de
Do you want to choose more(y/n)
n
The adjacency matrix is :
1 1 1 1 1
1 0 1 1 1
1 1 0 1 1
1 1 1 0 1
1 1 1 1 0
The given graph is complete graph

--------------------------------
Process exited after 40.14 seconds with return value 0
Press any key to continue . . .
```

Output 2

```
D:\prachi_c\Q10.exe
Enter the number of vertices
5
Enter the name of the vertices
abcde
Enter the pair of vertices for edges one by one
ab
Do you want to choose more(y/n)
y bc
Do you want to choose more(y/n)
y aa
Do you want to choose more(y/n)
y bb
Do you want to choose more(y/n)
y ce
Do you want to choose more(y/n)
y da
Do you want to choose more(y/n)
n
The adjacency matrix is :
1 1 0 1 0
1 1 1 0 0
0 1 0 0 1
1 0 0 0 0
0 0 1 0 0
The given graph is not the complete graph

--------------------------------
Process exited after 20.27 seconds with return value 0
Press any key to continue . . .
```

Q17) Write a Program to accept a directed graph G and compute the in-degree and out-degree of each vertex.

Ans)

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

void directed_in_out(char V[]);
int main()
{
	char v[20];
	directed_in_out(v);
	return 0;
}
void directed_in_out(char V[])
{
	char in[10],out[10];
	bool f=true;
	int num;
	char A[10][10];
	cout<<"Enter the number of vertices in the graph"<<endl;
	cin>>num;
	char a;
	cout<<"Enter the name of your vertices"<<endl;
	for(int i=0; i<num ;i++)
	{
		cin>>V[i];
	}
	for(int a=0; a<num; a++)
	{
		for(int b=0; b<num; b++)
```

```
                {
                        A[a][b]='0';
                }
        }


        cout << "Enter the pairs of vertices to form an edge in the direction of vertice1-->vertice2:" << endl;
    char ch;
        char R[1][2];
    int index1 =0, index2 =0;
    bool flag=true;
    for(int i=0 ; i<num; i++)
    {
        for(int j=0 ; j<num ;j++)
        {
                A[i][j]='0';
                }
        }
    while (flag == true)
    {
      int count = 0;
      char a, b;
      cin >> R[0][0];
      cin >> R[0][1];
      a = R[0][0];
      b = R[0][1];
      bool ff=false;
      for(int i=0; i<num ;i++)
      {
          for(int j=0; j<num ;j++){
```

```
            if(a==V[i] && b==V[j])
            {
                ff=true;
                break;
                    }
                }
            }
            if(ff==false)
            {
                cout<<"ERROR! Wrong pair of vertices"<<endl;
            }
for (int i = 0; i < num; i++)

{
   if (a == V[i])
   {
      index1 = i;
   }
   if (b == V[i])
   {
      index2 = i;
   }

}
A[index1][index2]='1';

cout << "Do you want to choose more (y/n)" << endl;
cin >> ch;
```

```
    if (ch == 'n')
    {
        flag = false;
    }
}
cout<<endl;
cout<<"The adjacency matrix is : "<<endl;
        for(int i=0; i<num; i++)
        {
                for(int j=0 ; j<num; j++)
                {
                        cout<<A[i][j]<<" ";
                }
                cout<<endl;
        }
        int sum_out=0;
        int sum_in=0;
        cout<<endl;
        cout<<"------------OUTDEGREES----------"<<endl;
        cout<<endl;
        for(int i=0; i<num ; i++)
        {
                cout<<"The outdegree of vertice "<<V[i]<<" is ";
                for(int j=0; j<num ; j++)
                {

                        if(A[i][j]=='1')
                        {
                                sum_out+=1;
```

```
                }
            }
            cout<<sum_out;
            cout<<endl;
            sum_out=0;
        }
        cout<<endl;
        cout<<"-----------INDEGREES-----------"<<endl;
        cout<<endl;
        for(int i=0; i<num; i++)
        {
            cout<<"The indegree of vertice "<<V[i]<<" are ";
            for(int j=0 ; j<num ;j++)
            {
                if(A[j][i]=='1')
                {
                    sum_in+=1;
                }
            }
            cout<<sum_in;
            cout<<endl;
            sum_in=0;
        }
}
```

Output

D:\prachi_c\Q17.exe

```
Enter the number of vertices in the graph
5
Enter the name of your vertices
abcde
Enter the pairs of vertices to form an edge in the direction of vertice1--
a b
Do you want to choose more (y/n)
y b c
Do you want to choose more (y/n)
y  a a
Do you want to choose more (y/n)
y c d
Do you want to choose more (y/n)
y d e
Do you want to choose more (y/n)
y ac
Do you want to choose more (y/n)
y e d
Do you want to choose more (y/n)
y b e
Do you want to choose more (y/n)
y e a
Do you want to choose more (y/n)
y e b
Do you want to choose more (y/n)
n

The adjacency matrix is :
1 1 1 0 0
0 0 1 0 1
0 0 0 1 0
0 0 0 0 1
1 1 0 1 0

------------OUTDEGREES----------

The outdegree of vertice a is 3
The outdegree of vertice b is 2
The outdegree of vertice c is 1
The outdegree of vertice d is 1
The outdegree of vertice e is 3

-----------INDEGREES-----------

The indegree of vertice a are 2
The indegree of vertice b are 2
The indegree of vertice c are 2
The indegree of vertice d are 2
The indegree of vertice e are 2
```

Q18) Given a graph G, Write a Program to find the number of paths of length n between the source and destination entered by the user.

Ans)

```
#include <iostream>
using namespace std;
void input();
void path(int A[][10], int num, char V[]);
int main()
{
   input();
   return 0;
}
void input()
{
   int A[10][10];
   int num;
   cout << "Enter the number of vertices" << endl;
   cin >> num;
   char V[num];
   cout << "Enter the name of your vertices" << endl;
   for (int i = 0; i < num; i++)
   {
      cin >> V[i];
   }
   int e_num;
   cout << "Enter the number of edges in your graph" << endl;
   cin >> e_num;
   char E[e_num][2];
   cout << "Enter the pair of edges" << endl;
```

```
bool ff;
for (int i = 0; i < e_num; i++)
{
    for (int j = 0; j < 2; j++)
    {
        cin >> E[i][j];
        for(int k=0; k<num; k++)
        {
            if (E[i][j] == V[k])
            {
                ff = true;
                break;
            }
            else{
                ff=false;
            }
        }
        if(ff==false)
        {
            cout<<"Wrong vertex input"<<endl;
            ff=true;
            i--;
            break;
        }
    }
}

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

```
    for (int j = 0; j < num; j++)
    {
        A[i][j] = 0;
    }
}
for (int i = 0; i < e_num; i++)
{
    int index1 = 0;
    int index2 = 0;
    for (int j = 0; j <= num; j++)
    {
        if (E[i][0] == V[j])
        {
            index1 = j;
        }
        if (E[i][1] == V[j])
        {
            index2 = j;
        }
    }
    A[index1][index2] = 1;
    A[index2][index1] = 1;
}
cout << "The adjacency matrix is: " << endl;
for (int i = 0; i < num; i++)
{
    for (int j = 0; j < num; j++)
    {
        cout << A[i][j] << " ";
```

```
        }
        cout << endl;
    }
    path(A, num, V);
}
void path(int A[][10], int num, char V[])
{
    int res[10][10];
    int sum = 0;
    int length;
    cout << "Enter the length of the path" << endl;
    cin >> length;
    char start, dest;
    cout << "Enter the starting vertex" << endl;
    cin >> start;
    cout << "Enter the destination vertex" << endl;
    cin >> dest;
    for (int i = 0; i < num; i++)
    {
        for (int j = 0; j < num; j++)
        {
            res[i][j] = 0;
        }
    }
    for (int i = 0; i < length; i++)
    {
        for (int j = 0; j < length; j++)
        {
            for (int k = 0; k < length; k++)
```

```
        {
            sum += A[i][k] * A[k][j];
        }
        res[i][j] = sum;
        sum = 0;
    }
}
cout << "The resultant matrix is: " << endl;
for (int i = 0; i < num; i++)
{
    for (int j = 0; j < num; j++)
    {
        cout << res[i][j] << " ";
    }
    cout << endl;
}
int i1, i2;
for (int j = 0; j <= num; j++)
{
    if (start == V[j])
    {
        i1 = j;
    }
    if (dest == V[j])
    {
        i2 = j;
    }
}
int l;
```

```
    if (res[i1][i2] != 0)
    {
        l = res[i1][i2];
        cout << "The total available path are " << endl;
        cout << l << endl;
    }
    else
    {
        cout << "There are no such path available" << endl;
    }

    return;
}
```

Output

```
D:\prachi_c\Q18.exe
ac
ad
bd
be
cd
de
The adjacency matrix is:
0 1 1 1 0
1 0 0 1 1
1 0 0 1 0
1 1 1 0 1
0 1 0 1 0
Enter the length of the path
2
Enter the starting vertex
a
Enter the destination vertex
b
The resultant matrix is:
2 2 2 0 0
2 3 4 0 0
4 5 6 0 0
0 0 0 0 0
0 0 0 0 0
The total available path are
2

--------------------------------
Process exited after 45.47 seconds with return value 0
Press any key to continue . . .
```

Q19) Given an adjacency matrix of a graph, write a program to check whether a given set of vertices {v_1,v_2,v_3.....,v_k} forms an Euler path / Euler Circuit (for circuit assume v_k=v_1).

Ans)

```
#include <iostream>
using namespace std;
void input();
void euler(int A[][10],int num);
int main()
{
   input();
}
void input()
{
   int A[10][10];
   int num;
   cout << "Enter the number of vertices" << endl;
   cin >> num;
   char V[num];
   cout << "Enter the name of your vertices" << endl;
   for (int i = 0; i < num; i++)
   {
      cin >> V[i];
   }
   int e_num;
   cout << "Enter the number of edges in your graph" << endl;
   cin >> e_num;
   char E[e_num][2];
   bool ff;
```

```
cout << "Enter the pair of edges" << endl;
for (int i = 0; i < e_num; i++)
{
   for (int j = 0; j < 2; j++)
   {
      cin >> E[i][j];
      for(int k=0; k<num;k++)
      {
       if(E[i][j]==V[k])
       {
                        ff=true;
                                break;
                }
                else{
                        ff=false;
                }
        }
        if(ff=false)
        {
                cout<<"Wrong vertex input"<<endl;
                ff=true;
                i--;
                break;
        }
   }
}
for (int i = 0; i < num; i++)
{
   for (int j = 0; j < num; j++)
```

```
    {
        A[i][j] = 0;
    }
}
for (int i = 0; i < e_num; i++)
{
    int index1 = 0;
    int index2 = 0;
    for (int j = 0; j <= num; j++)
    {
        if (E[i][0] == V[j])
        {
            index1 = j;
        }
        if (E[i][1] == V[j])
        {
            index2 = j;
        }
    }
    A[index1][index2] = 1;
    A[index2][index1] = 1;
}
cout << "The adjacency matrix is: " << endl;
for (int i = 0; i < num; i++)
{
    for (int j = 0; j < num; j++)
    {
        cout << A[i][j] << " ";
    }
```

```
        cout << endl;
    }
    euler(A,num);
}
void euler(int A[][10],int num)
{
    int count=0;
    int deg=0;
    for(int i=0; i<num; i++)
    {
        for(int j=0; j<num; j++)
        {
            if(A[i][j]==1)
            {
                deg+=1;
            }
        }
        if(deg%2==0)
        {
            count++;
        }
        deg=0;
    }
    if(count==num)
    {
        cout<<"The given graph is Euler circuit"<<endl;
    }
    else{
        if(count==num-2)
```

```
        {
            cout<<"The given graph is euler path"<<endl;
        }
        else{
            cout<<"The given graph is neither euler circuit nor euler path"<<endl;
        }
    }
}
```

Output1

```
D:\prachi_c\euler.exe
Enter the number of vertices
6
Enter the name of your vertices
abcdef
Enter the number of edges in your graph
8
Enter the pair of edges
ab af bc cf ce cd ed fe
The adjacency matrix is:
0 1 0 0 0 1
1 0 1 0 0 0
0 1 0 1 1 1
0 0 1 0 1 0
0 0 1 1 0 1
1 0 1 0 1 0
The given graph is euler path

--------------------------------
Process exited after 38.41 seconds with return value 0
Press any key to continue . . .
```

Output2

D:\prachi_c\euler.exe

```
Enter the number of vertices
6
Enter the name of your vertices
abcdef
Enter the number of edges in your graph
9
Enter the pair of edges
ab ac ad ae de ef ce bc cf
The adjacency matrix is:
0 1 1 1 1 0
1 0 1 0 0 0
1 1 0 0 1 1
1 0 0 0 1 0
1 0 1 1 0 1
0 0 1 0 1 0
The given graph is Euler circuit

--------------------------------
Process exited after 33.18 seconds with return value 0
Press any key to continue . . .
```

Q20) Given a full `m-ary` tree with `i` internal vertices, Write a Program to find the number of leaf nodes.

Ans)

```
#include<iostream>
using namespace std;
int resultLeaves(int internal,int arry)
{
   int sum=0;
   sum = internal*(arry-1)+1;
   return sum;
}
int main()
{
   int m_arry;
   int nodes;
   cout<<"Enter the number of internal nodes in the tree"<<endl;
   cin>>nodes;
   cout<<"The number of children each internal node have (m_arry)"<<endl;
   cin>>m_arry;
   int leaves;
   leaves=resultLeaves(nodes,m_arry);
   cout<<"Total number of leaves in the tree are"<<endl;
   cout<<leaves<<endl;
   return 0 ;
}
```

Output

D:\prachi_c\Q20.exe

```
Enter the number of internal nodes in the tree
5
The number of children each internal node have (m_arry)
3
Total number of leaves in the tree are
11

--------------------------------
Process exited after 5.192 seconds with return value 0
Press any key to continue . . .
```