Ass. 1 ** Program for Matrix Addition ***

```
#include<iostream.h>
#include<conio.h>
void main()
{
clrscr();
int i,j, a[2][2],b[2][2],c[2][2],m,n,p,q;
cout<<"enter the row and column size of A matrix\n";
cin>>m>>n;
cout<<"\n enter the row and column size of B matrix\n";
cin>>p>>q;
if((m==p) && (n==q))
{
cout<<"matrices can be added or subtracted...\n";
cout<<"enter matrix A elements..\n";
for(i=0;i<m;i++)
{
for(j=0;j<n;j++)
{
cin>>a[i][j];
}
}
cout<<"enter matrix B element..\n";
for(i=0;i<p;i++)
{
for(j=0;j<q;j++)
{
cin>>b[i][j];
}
}
cout<<"Addition of matrix A and B\n" ;
for(i=0;i<m;i++)
{
for(j=0;j<n;j++)
{
c[i][j]=a[i][j]+b[i][j];
}
}
cout<<"Sum of A and B\n";
for(i=0;i<m;i++)
{
for(j=0;j<n;j++)
{
cout<<c[i][j]<<"";
cout<<endl;
} } } }
```

***** OUT PUT*****

```
enter the row and column size of A matrix
2
2
 enter the row and column size of B matrix
2
2
matrices can be added or subtracted...
enter matrix A elements..
2  2
4  5
enter matrix B element..
4  5
4  5
Addition of matrix A and B
Sum of A and B
6   7
8   10
```

Ass. 2 *** PROGRAM FOR STACK ******

```
#include<iostream.h>
#include<conio.h>
#include<process.h>
#define size 5
int s[size], ele, top=-1,i,n;
class stack
{
public:
void push()
{
        if(top==size-1)
        {
        cout<<"stack is overflow\n";
        return;
        }
        cout<<"enter the element to be inserted\n";
        cin>>ele;
        top=top+1;
        s[top]=ele;
}
void pop()
{
        if(top==-1)
        {
        cout<<"stack is underflow\n";
        return;
        }
        ele=s[top];
        cout<<ele<<"\t is deleted\n" ;
        top=top-1;
}
void display()
{
if(top==-1)
{
cout<<"stack is empty\n";
return;
}
cout<<"elements of stack are:-\n";
for(i=top;i>=0;i--)
{
                cout<<s[i];
 }  } };
void main()
{
stack s;
int ch;
clrscr();
while(ch!=4)
{
cout<<"\n\t stack operation";
cout<<"\n\t1. push operation";
cout<<"\n\t2. pop operation";
cout<<"\n\t3. display operation";
cout<<"\n\t4. exit operation";
cout<<"\nenter the choice\t";
cin>>ch ;
if(ch==1)
s.push();
if(ch==2)
s.pop();
if(ch==3)
s.display();
}}
```

*****OUTPUT*****

```
    stack operation
       1. push operation
       2. pop operation
       3. display operation
       4. exit operation
enter the choice       1
enter the element to be inserted
10
    stack operation
       1. push operation
       2. pop operation
       3. display operation
       4. exit operation
enter the choice       1
enter the element to be inserted
20
    stack operation
       1. push operation
       2. pop operation
       3. display operation
       4. exit operation
enter the choice       1
enter the element to be inserted
30
  stack operation
       1. push operation
       2. pop operation
       3. display operation
       4. exit operation
enter the choice       3

10 20 30
```

Ass . 3 **** PROGRAM FOR QUEUE ****

```
#include<iostream.h>
#include<conio.h>
#include<process.h>
#define SIZE 10
static int front=0;
static int end=-1;
class queue
{
  private:
     int ar[SIZE];
  public:
     void insert(int item);
     void delque();
     void viewque();
};
void queue::insert(int item)
{
   if(end==SIZE-1)
     cout<<"\nThe Queue is Full!!!";
   else
    { ar[++end]=item;
      cout<<"\nElement succesfully inserted in the Queue!!!";
    }
}
void queue::delque()
{
  if(end<0)
    cout<<"\nQueue Under flow!!!";
  else
   { front++;
     cout<<"\nElement sucessfully deleted from the Queue!!!";
   }
}
void queue::viewque()
{ if(end<0)
    cout<<"\nThe Queue is Empty it cannot be Viewed!!!";
  else
    for(int i=front;i<=end;i++)
      cout<<ar[i]<<" ";
}
void main()
{
  clrscr();
  char choice;
  int ch, num;
  queue ob;
  do
  {
  clrscr();
  cout<<"\n\n\t\t\tQ U E U E  O P E R A T I O N S";
  cout<<"\n\t\t\t-----------------------------";
  cout<<"\n\n1.INSERT";
  cout<<"\n2.DELETE";
  cout<<"\n3.DISPLAY";
  cout<<"\n4.EXIT";
  cout<<"\n\nEnter your choice:";
  cin>>ch;
   switch(ch)
   {
     case 1: cout<<"\nEnter the Element you want to Push:";
             cin>>num;
             ob.insert(num);
              break;
     case 2: ob.delque(); break;
     case 3: ob.viewque(); break;
     case 4: exit(0);
     default: cout<<"\nPlease Enter a Valid Choice(1-4)!!!";
   }
  cout<<"\nDo you want to Continue(Y/N):";
  cin>>choice;
  }while(choice=='y' || choice=='Y');
getch();
}
```

***** OUT PUT *****

```
  Q U E U E  O P E R A T I O N S
  ------------------------------
1.INSERT
2.DELETE
3.DISPLAY
4.EXIT

Enter your choice:3
10 20 30 40
Do you want to Continue(Y/N):y
```

Ass. 4 ***Single Linked List ***

```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
struct node
{
        int info;
        struct node *link;
};
struct node *first, *newnode, *temp, *parent;
int totnodes,n;

void addbegin()
{
newnode=new (struct node);
newnode->link=NULL;
if(newnode==NULL)
{
cout<<"\nlist is full" ;
return ;
}
cout<<"\nenter the info of newnode\n" ;
cin>>newnode->info;
if(first==NULL)
{
first=newnode;
}
else
{
newnode->link=first;
first=newnode;
}
totnodes++;
}

void addlast()
{
newnode=new(struct node);
newnode->link=NULL;
if(newnode==NULL)
{
cout<<"\nlist is full" ;
return;
}
cout<<"\nenter info for newnode" ;
cin>>newnode->info;
temp=first;
while(temp->link!=NULL)
{
temp=temp->link;
}
if(first==NULL)
{
first=newnode;
}
else
{
temp->link=newnode;
}
totnodes++;
}

void display()
{
if(first==NULL)
{
cout<<"\nlist is empty";
return;
}
temp=first;
cout<<"\nelement of list are" ;
while(temp!=NULL)
{
cout<<"\n"<<temp->info;
temp=temp->link;
} }

void delfirst()
{
if(first==NULL)
{
cout<<"\nlist is empty" ;
return;
}
temp=first;
first=first->link;
cout<<temp->info<<"\tis deleted";
free(temp);
totnodes--;
}

void dellast()
{
if(first==NULL)
{
cout<<"\nlist is empty";
return;
```

```
}
temp=first;
while(temp->link!=NULL)
{
parent=temp;
temp=temp->link;
}
if(temp==first)
{
first=NULL;
}
else
{
parent->link=NULL;
cout<<temp->info<<"\t is deleted" ;
free(temp);
totnodes--;
}}
```

void addnpos()

```
{
int c;
newnode=new(struct node);
newnode->link=NULL;
if(newnode==NULL)
{
cout<<"\nlist is full";
return;
}
cout<<"\nenter the position for inserting the node";
cin>>n;
if(n>totnodes)
{
cout<<"\nInvalid node position" ;
return;
}
cout<<"\nenter the information for a newnode";
cin>>newnode->info;
temp=first;
c=1;
while(c<n)
{
parent=temp;
temp=temp->link;
c++;
}
parent->link=newnode;
newnode->link=temp;
totnodes++;
}
```

void delnpos()

```
{
int c;
if(first==NULL)
{
cout<<"\nlist is empty";
return;
}
cout<<"\nenter the node position to be deleted" ;
cin>>n;
if(n>totnodes)
{
cout<<"\ninvalid node position";
return;
}
temp=first;
c=1;
while(c<n)
{
parent=temp;
temp=temp->link;
c++;
}
parent->link=temp->link;
cout<<temp->info<<"\tis deleted" ;
free(temp);
totnodes--;
}
```

void main()

```
{
int ch;
clrscr();
first=NULL;
totnodes=0;
ch=1;
while(ch!=8)
{
cout<<"\n\t---------
**********SINGLY LINKED LIST**********----------" ;
cout<<"\n1.ADDBEGIN ";
cout<<"\n2. ADDLAST";
cout<<"\n3. ADDNPOS";
cout<<"\n4. DELFIRST";
cout<<"\n5.DELLAST";
cout<<"\n6. DELNPOS";
cout<<"\n7. DISPLAY";
```

```
cout<<"\n8.EXIT";
cout<<"\nenter your choice\n";
cin>>ch;
if(ch==1)
addbegin();
if(ch==2)
addlast();
if(ch==3)
addnpos();
if(ch==4)
delfirst();
if(ch==5)
dellast();
if(ch==6)
delnpos();
if(ch==7)
display();
}}
```

********* OUT PUT **************

```
 **SINGLY LINKED LIST **
1.ADDBEGIN
2. ADDLAST
3. ADDNPOS
4. DELFIRST
5.DELLAST
6. DELNPOS
7. DISPLAY
8.EXIT
enter your choice
1
enter the info of newnode
20
enter your choice
 1
enter the info of newnode
10
enter your choice
1
enter the info of newnode
30

enter your choice
7
element of list are
30
10
20
```

Ass. 5 ** PROGRAM FOR BUBBLE SORT ***

```
#include<iostream.h>
#include<conio.h>
class demo
{
        int i,arr[20],len,j;
        public:
                void get();
                void put();
                void sort();
};
        void demo :: get()
{
        cout<<"\n\n\n\n\n How Elements You want to Insert\n\n\n\n";
        cin>>len;
        cout<<"\n\n\n\nNow Insert Total length Element\n\n\n";
        for(int i=1;i<=len;i++)
        {
        cin>>arr[i];
        }
}
        void demo::put()
        {
        cout<<"\n\n\n Element are\n\n
";
        for (i=1;i<=len;i++)
        {
        cout<<arr[i]<<"\n";
        }
}
        void demo ::sort()
{
        for (int i=1;i<=len-1;i++)
        {
        for(int j=1;j<=len-i;j++)
        {
        if(arr[j]>arr[j+1])
        {
        int temp;
        temp=arr[j+1];
        arr[j+1]=arr[j];
        arr[j]=temp;
        } } }
}
        void main()
        {
        clrscr();
        demo d;
        d.get();
        d.put();
        d.sort();
        d.put();
        getch();        }
```

*******OUTPUT*******

```
 How Elements You want to Insert
5
Now Insert Total length Element
32
51
27
85
66
Elements are
27
32
51
66
85
```

Ass. 6 ** PROGRAM FOR RECURSION***

```
#include<iostream.h>
#include<conio.h>
unsigned long fact(int n)
{
        unsigned long int fact=1,n;
       if(n>1)
       {
              fact=fact*fact(n-1);
       }
       return fact;
}
void main()
{

int n;
unsigned long int fact;
cout<<"Enter the number";
cin>>n;
fact=fact(n);
cout<<"Factorial of n is="<<fact;
}
```

****OUTPUT****

```
Enter the number
5
Factorial of n is= 120
```

Ass. 7 ** PROGRAM FOR LINEAR SEARCH**

```
#include<iostream.h>
#include<conio.h>
#define N 5
 class linear
 {

 private: int a[N],i,ele;
 public:
        get();
        seq();
 };
linear :: get()
{
cout<<"enter the array element\n";
for(i=0;i<N;i++)
cin>>a[i];
}
linear :: seq()
{
int c;
c=0;
cout<<"enter the element to be searched\n";
cin>>ele;
for(i=0;i<N;i++)
{
        if(a[i]==ele)
        {
                c++;
        }
}
        if(c==0)
        {
        cout<<ele<<"not found\n";
        }
        else
        {
        cout<<ele<<"\tfound\t"<<c<<"\ttimes\n" ;
        }
}
void main()
{
clrscr();
linear L;
L.get();
L.seq();
}
```

OUTPUT :--

Enter the Array Element
10
20
10
30
40
enter the element to be searched
10
10 found 2 times

Ass. 8 *** Program for Binary Search ****

```
#include<iostream.h>
#include<conio.h>
#define N 5
class Binary
{
private:
int  a[N],i,first,last,mid,flag,ele;
public:
        void getdata();
        void bsearch();
        void putdata();
};
void Binary::getdata()
{
cout<<"enter the array element\n";
for(i=0;i<N;i++)
cin>>a[i];
}
void Binary::bsearch()
{
cout<<"enter the element to be search\n";
cin>>ele;
first=0;
last=N-1;
flag=0;
while(first<=last)
{
mid=(first+last)/2;
if(a[mid]<ele)
{
first=mid+1;
}
else
{
if(a[mid]>ele)
{
last=mid-1;
}
else
{
cout<<ele<<"\t"<<"found at"<<"\t"<<mid<<"\t"<<"position";
flag=1;
return;
} } } }
void Binary::putdata()
{
if(flag==0)
{
cout<<ele<<"\t"<<"Not Found";
}}
void main()
{
clrscr();
Binary b;
b.getdata();
b.bsearch();
b.putdata();
getch();
}
```

****** OUT PUT *******

```
Enter The Array Element
10 20 30 40 50
Enter the element to be searched
50
50 found at 4 position
```

Ass. 9 ** PPROGRAM FOR TOWER OF HANOI **

```
#include <iostream.h>
#include <conio.h>
void tower(int a,char from,char aux,char to)
{
   if(a==1)
   {
     cout<<"\t\tMove disc 1 from "<<from<<" to "<<to<<"\n";
     return;
   }
   else
   {
     tower(a-1,from,to,aux);
     cout<<"\t\tMove disc "<<a<<" from "<<from<<" to "<<to<<"\n";
     tower(a-1,aux,from,to);
   }
}
void main()
{
    clrscr();
    int n;
    cout<<"\n\t\t*****Tower of Hanoi*****\n";
    cout<<"\t\tEnter number of discs : ";
    cin>>n;
    cout<<"\n\n";
    tower(n,'A','B','C');
    getch();
}
```

*** OUT PUT ***

```
*****Tower of Hanoi*****
     Enter number of discs : 3

     Move disc 1 from A to C
     Move disc 2 from A to B
     Move disc 1 from C to B
     Move disc 3 from A to C
     Move disc 1 from B to A
     Move disc 2 from B to C
     Move disc 1 from A to C
```