Name: Sanehal V. Nikam
Roll No.: 108

Program: Adjacency Matrix Represent of Graph

-----------------\PROGRAM\--------------------

```
#include<iostream.h>

#include<conio.h>

#include<stdio.h>

#include<process.h>

class ADM

{

      int n,i,j,adm[10][10],node[5];

public:

      void Insert();

      void Show();

};

void ADM :: Insert()

{

       cout<<"\n How many nodes are to be inserted for the graph
:"<<endl;

       cin>>n;

       cout<<"\n Enter\t"<<n<<"nodes into the graph: "<<endl;

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

       {

            cin>>node[i];

       }

       cout<<"\n Enter the adjacency matrix for graph : "<<endl;

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

       {

            for(j=0;j<n;j++)

            {
                cout<<"\t a["<<i<<"]["<<j<<"] :";
                cin>>adm[i][j];
            }
         }
}
void ADM :: Show()
{
      cout<<"\n Adjacency Matrix is : "<<endl;
      for(i=0;i<n;i++)
      {
            for(j=0;j<n;j++)
            {
                  cout<<"\t"<<adm[i][j];
            }
            cout<<endl;
      }
}
void main()
{
      ADM a;
      int ch;
      clrscr();
      a.Insert();
      a.Show();
      getch();


}
```

------------------\OUTPUT\------------------

```
How many modes are to be inserted for the graph:

3

Enter the adjacency matrix for graph:

a[0][0]:1

a[0][1]:2

a[0][2]:4

a[0][0]:5

a[0][1]:2

a[0][2]:4

a[0][0]:5

a[0][1]:6

a[0][2]:7

Adjacency Matrix is:

1        2     4

5        2     4

5        6      7
```

Program: Binary Search

----------------\PROGRAM\-------------------

```
#include<iostream.h>

#include<conio.h>

class Array

{

      int A[10],N;

      int k,item;

      public:

            void Get();

            void Sort();

            void BSearch();

};

void Array :: Get()

{

      cout << "Enetr N:";

      cin >> N;


      cout << "Enter element :";

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

      {

            cin >> A[i];

      }

}

void Array :: Sort()

{

      int temp;

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

      {

              for(int j=i+1;j<N;j++)

              {

                  if(A[i] > A[j])

                  {
                      temp=A[i];
                      A[i]=A[j];
                      A[j]=temp;
                  }
              }
      }
      cout << "\n Elements after sorting :\n";
      for(i=0;i<N;i++)
      {
            cout << A[i] << "\t";
      }
}
void Array :: BSearch()
{
      int Beg=0, End=N;
      int Mid = (Beg+End)/2;
      int Loc;

      cout << "\n Enter element to search : ";
      cin >> item;

      while(Beg <= End && A[Mid] != item)
      {
            if(item < A[Mid])
            {
                  End = Mid-1;
            }
            else
            {
                  Beg = Mid+1;
            }
            Mid = (Beg+End)/2;
```

```
      }
      if(A[Mid] == item)
      {
            Loc=Mid;
            cout << "Item is on " << Loc+1 << " Location\n\n";
      }
      else
      {
            Loc=NULL;
            cout << "Element is not found";
      }
}
void main()
{
      clrscr();
      Array a;
      a.Get();
      a.Sort();
      a.BSearch();
      getch();
}
```

------------------\OUTPUT\------------------

```
Enter N: 5
Enter the element: 11
33
22
44
55
Element after sorting:
11      22     33      44      55


Enter Element to search: 33
Item is on 3 location
```

Program Name : Bubble Sort

----------------\PROGRAM\-------------------

```
#include<iostream.h>

#include<conio.h>

class sorting

{

private:

  int b[10], no, temp, i, j;

public:

  void in();

  void sort();

  void out();

};

void sorting ::in()

{

  cout << "Enter Size of an array : \n";

  cin >> no;

  cout << "Enter array elements : \n";

  for (i = 0; i < no; i++)

  {

    cin >> b[i];

  }

}

void sorting ::sort()

{


  for (i = 0; i < no; i++)

  {

    for (j = 0; j < no; j++)

    {

      if (b[i] < b[j])

      {

        temp = b[i];
        b[i] = b[j];
        b[j] = temp;
      }
    }
  }
}
void sorting ::out()
{
  cout << "After sorting array elements are : \n";
  for (i = 0; i < no; i++)
  {
    cout << b[i] << "\t";
  }
}
void main()
{
  clrscr();
  sorting obj;
  obj.in();
  obj.sort();
  obj.out();
  getch();
}
```

Output :

```
                Enter Size of an array :5
                Enter array elements :9 8 5 2 1
                sorting array elements are :
                1       2       5       8       9
```

Program: Linear Search

---------------\PROGRAM\------------------

```
#include<iostream.h>

#include<conio.h>

class Array

{

      int A[10],N;

      int k,item;

      public:

            void Get();

            void LSearch();

};

void Array :: Get()

{

      cout << "Enter the size of array :";

      cin >> N;


      cout << "Enter elements :";

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

      {

            cin >> A[i];

      }

}

void Array :: LSearch()

{

      cout << "\n Enter item:";

      cin >> item;


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

      {

            if(A[i]==item)

            {
                cout << "Element " <<item << " is found at " << i+1 
<<" location\n\n";
           }
      }
}
void main()
{
      clrscr();
      Array a;
      a.Get();
      a.LSearch();
      getch();
}
```

-------------------\OUTPUT\------------------

```
Enter the size of Array: 4
Enter the element: 11
33
22
44
Enter item: 22
Element 22 is found at 3 Location
```

Pratical Name : Matrix program

-----------------\PROGRAM\-------------------

Program :

```
#include <iostream.h>
#include <conio.h>
#include <process.h>
class mat
{
private:
    int a[3][3], b[3][3], c[3][3], i, j, k;
public:
    void get();
    void ADD();
    void SUB();
    void MUL();
    void DIV();
};
void mat ::get()
{
    cout << "Enter Elements for first Matrix : " << endl;
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            cin >> a[i][j];
        }
    }
    cout << "Enter Elements for secomd Matrix : " << endl;
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            cin >> b[i][j];
        }
    }
}
void mat ::ADD()
{
    cout << "Addition of Matrix is " << endl;
    for (i = 0; i < 3; i++)
    {
        for ( j = 0; j < 3; j++)
        {
            c[i][j] = a[i][j] + b[i][j];
           cout << c[i][j] << "\t";
        }
        cout<<"\n";
    }
}
void mat ::SUB()
{
    cout << "Substractinon of Matrix is " << endl;
    for ( i = 0; i < 3; i++)
    {
        for ( j = 0; j < 3; j++)
        {
            c[i][j] = a[i][j] - b[i][j];
            cout << c[i][j] << "\t";
        }
        cout<<"\n";
    }
}
void mat ::MUL()
{
    cout << "Multiplication of Matrix is " << endl;
    for ( i = 0; i < 3; i++)
    {
        for ( j = 0; j < 3; j++)
        {
            c[i][j] = 0;
            for ( k = 0; k < 3; k++)
            {
                c[i][j] = c[i][j] + a[i][k] * b[k][j];
                cout << c[i][j] << "\t";
            }
        }
        cout<<"\n";
    }
}
void mat :: DIV()
{
    cout<< "Division of Matrix is " << endl;
    for ( i = 0; i < 3; i++)
    {
        for ( j = 0; j < 3; j++)
        {
            c[i][j] = a[i][j] / b[i][j];
            cout << c[i][j] << "\t";
        }
         cout<<"\n";
    }
}
void main()
{
    clrscr();
    mat o;
```

```
int d;
o.get();
do{
    cout << "1.Addition" << endl;
    cout << "2.Substractinon" << endl;
    cout << "3.Multiplication" << endl;
    cout << "4.Division" << endl;
    cout << "Exit"<< endl;
cout << "Enter your choice" << endl;
    cin >> d;
    switch (d)
    {
    case 1:
        o.ADD();
        break;
    case 2:
        o.SUB();
        break;
    case 3:
        o.MUL();
        break;
    case 4:
        o.DIV();
        break;
    case 5:
        exit(0);
        break;
    default:
        cout << "Invalid Choice !!" << endl;
    }
}while(d<=5);
getch();}
```

Practical Name: RECURSION

-----------------\PROGRAM\--------------------

```
#include<iostream.h>

#include<conio.h>

 class Fact
  {
   public:
      long Facto(long n);
  };

long Fact::Facto(long n)
 {
   if(n==1)
     return 1;
   else
     {
     long a=n;
     n--;
     return(a*Facto(n));
     }
 }
 void main()
 {
   long x;
   clrscr();
   cout<< "Enter the Number: ";
   cin>>x;
   Fact f;
   long n=f.Facto(x);
   cout<<"\n Factorial:"<<n;
   getch();
 }
```

------------------\OUTPUT\-----------------

```
Enter the Number: 5

factorial: 120
```

Program: Queue

-----------------\PROGRAM\--------------------

```
#include<iostream.h>
#include<conio.h>
class Queue
{
      int a[4],R,F;
      public:
      Queue()
      {
            F=R=0;
      }

      void Insert();
      void Delete();
      void Show();
};
void Queue :: Insert()
{
      int item;

      if(R>=3)
      {
            cout << "\n Overflow";
      }
      else
      {
            cout << "\n enter item :";
            cin >> item;

            R++;
            a[R]=item;
```

```
                if(F==0)
                {
                        F=1;
                }
        }
}
void Queue :: Delete()
{
        if(F==0)
        {
                cout << "\n Underflow";
        }
        else
        {
                cout << "\n Deleted element is :" << a[F];

                if(F==R)
                {
                        F=R=0;
                }
                else
                {
                        F++;
                }

        }
}
void Queue :: Show()
{
        if(F==0)
        {
            cout << "\n Empty.";
      }
      else
      {
                     cout << "\n Elements are : \n";
            for(int i=F;i<=R;i++)
            {
                  cout << "\n" << a[i];
            }
      }
}
void main()
{
      Queue s;
      int ch;
      clrscr();
            cout << "\n--------------------------------";
      cout << "\n 1. INSERT";
      cout << "\n 2. DELETE";
      cout << "\n 3. SHOW";
      cout << "\n 4. EXIT";
      cout << "\n--------------------------------";
      do
      {
            cout << "\n Enter choice : ";
            cin >> ch;
            switch(ch)
            {
                  case 1: s.Insert();
                        break;
                  case 2: s.Delete();
                        break;
```

```
                case 3: s.Show();

                     break;

                default: cout << "\n Wrong Choice.";

           }

      }while(ch<=3);

      getch();

}
```

------------------\OUTPUT\------------------

```
1.Insert
2.Delete
3.show
4.exit

Enter the choice:1
Enter  item:11

Enter the choice:1
Enter  item:22

Enter the choice:1
Enter  item:33

Enter the choice:2
Delete element is :11

Enter the choice:3
Element are:
22
33
```

Program: SINGLY LINKED LIST

-----------------\PROGRAM\--------------------

```
#include<iostream.h>
#include<conio.h>
int cnt=1;
class LinkList
{
      int Info,loc;
      LinkList *Link;
public:
      void Insert();
      void Delete();
      void Display();
};
LinkList *Start,*Node,*Temp1,*Temp2;
void LinkList :: Insert()
{
      int item;
      cout << "\n At which location u want to insert item:";
      cin >> loc;
      if(loc >  cnt)
      {
            cout << "\n Invalid Position.";
            getch();
            return;
      }
      Node = new LinkList;

      cout << "\n Enter the item";
      cin >> item;
      Node->Info = item;
      Node->Link = NULL;
```

```
if(loc == 1)
{
      if(Start == NULL)
      {
            Start=Node;
            cnt++;
            return;
      }
      else
      {
            Node->Link=Start;
            Start=Node;
            cnt++;
            return;
      }
}
else
{
      Temp1=Start;
      Temp2=Start;
      for(int i=1; i<loc-1; i++)
            Temp1=Temp1->Link;
      if(Temp1->Link==NULL)
      {
            Temp1->Link=Node;
            cnt++;
            return;
      }
      else
      {
            for(int i=1; i<loc; i++)
            Temp2=Temp2->Link;
```

```
                Temp1->Link=Node;
                Node->Link=Temp2;
                cnt++;
                return;
            }
        }
    }

    void LinkList :: Delete()
    {
        int loc;

        if(Start == NULL)
        {
            cout << "\n Link list is empty.";
            return;
        }
        cout << endl << "Enter the Location of Node to be deleted : ";
        cin >> loc;
        Temp1=Start;
        Temp2=Start;
        if(loc >= cnt)
        {
            cout << "Invalid Position.";
            return;
        }
        if(loc==1)
        {
            Start=Start->Link;
            cnt--;
            return;
        }
        for(int i=1;i<loc-1;i++)
                Temp1=Temp1->Link;
        if(Temp1->Link->Link==NULL)
        {
                Temp1->Link=NULL;
                cnt--;
                return;
        }
        else
        {
                for (i=1;i<loc+1;i++)
                Temp2=Temp2->Link;
                Temp1->Link=Temp2;
                cnt--;
                return;
        }
}

void LinkList :: Display()
{
        if(Start==NULL)
        {
        cout << "\n There are no elements in list.";
        }
        else
        {
                for(Node=Start;Node!=NULL;Node=Node->Link)
                cout << Node->Info << endl;
        }
}
void main()
{
	LinkList s;
	int ch;
	clrscr();
	cout << "\n CHOICES FOR DOUBLY LINKED LIST.....";
	cout << "\n -------------------------------";
	cout << "\n 1. INSERT";
	cout << "\n 2. DELETE";
	cout << "\n 3. DISPLAY";
	cout << "\n 4. Exit";
	cout << "\n -------------------------------";
	do
	{
		cout << "\n Enter choice : ";
		cin >> ch;
		switch(ch)
		{
			case 1: s.Insert();
				break;
			case 2: s.Delete();
				break;
			case 3: s.Display();
				break;
			case 4: Exit
				break;
			default: cout << "\n Wrong Choice.";
		}
	}while(ch<=4);
}
```

-------------------\OUTPUT\------------------

```
Choice for Single Linked list…..

1.Insert

2.Delete

3.Display
```

```
4.Exit

Enter choice:1
At which location u want to Insert item:1
Enter the item 11
Enter choice:1
At which location u want to Insert item:2
Enter the item 22
Enter choice:1
At which location u want to Insert item:3
Enter the item 33
Enter choice:2
At which location u want to Delete item:2
Enter choice:3
There are no elements in list :
11
33
```

Program: STACK.

-----------------\PROGRAM\--------------------

```
#include<iostream.h>

#include<conio.h>

class Stack

{

      int a[4],top;


      public:

      Stack()

      {

            top=0;

      }


      void Push();

      void Pop();

      void Show();

};

void Stack :: Push()

{

      int item;


      if(top==3)

      {

            cout << "\n Overflow";

      }

      else

      {

            cout << "\n enter item :";
```

```
          cin >> a[++top];
     }
}
void Stack :: Pop()
{
     if(top==0)
     {
          cout << "\n Underflow";
     }
     else
     {
          cout << "\n Deleted element is :" << a[top];
          top--;
     }
}
void Stack :: Show()
{
     if(top==0)
     {
          cout << "\n Empty.";
     }
     else
     {
          cout << "\n Elements are : \n";
          for(int i=top;i>=1;i--)
          {
               cout << "\n" << a[i];
          }
     }
```

```
}
void main()
{
    Stack s;
    int ch;
    clrscr();

    cout << "\n--------------------------------";
    cout << "\n 1. PUSH";
    cout << "\n 2. POP";
    cout << "\n 3. SHOW";
    cout << "\n 4. EXIT";
    cout << "\n--------------------------------";
    do
    {
        cout << "\n Enter choice : ";
        cin >> ch;
        switch(ch)
        {
            case 1: s.Push();
                break;
            case 2: s.Pop();
                break;
            case 3: s.Show();
                break;
            default: cout << "\n Wrong Choice.";
        }
    }while(ch<=3);
}
```

-----------------\OUTPUT\-----------------

```
1.push
2.pop
3.show
4.exit

Enter the choice:1
Enter  item:11

Enter the choice:1
Enter  item:22

Enter the choice:1
Enter  item:33

Enter the choice:2
Delete element is :33

Enter the choice:3
Element are:
11
22
```

Program: Tower of Hanoi

-----------------\PROGRAM\--------------------

```
#include<iostream.h>
#include<conio.h>
class tower
{
public:
void TOH();
};
void TOH(int n, char A, char B, char C)
{
if(n==1)
{
cout<<"\nShift top disk from tower"<<A<<" to tower"<<B;
return;
}
TOH(n-1,A,C,B);
cout<<"\nShift top disk from tower"<<A<<" to tower"<<B;
TOH(n-1,C,B,A);
}
void main()
{
int disk;
clrscr();
cout<<"Enter the number of disks:";
cin>>disk;
if(disk<1)
{
cout<<"There are no disks to shift";
}
else

{
```

```
cout<<"There are "<<disk<<"disks in tower 1\n";

TOH(disk, '1','2','3');

cout<<"\n\n"<<disk<<"disks in tower 1 are shifted to tower 2";

}

getch();


}
```

------------------\OUTPUT\-----------------

```
Enter the number of disk:3

There are 3 disks in tower 1


Shift top disk from tower 1 to tower 2

Shift top disk from tower 1 to tower 3

Shift top disk from tower 2 to tower 3

Shift top disk from tower 1 to tower 2

Shift top disk from tower 3 to tower 2

Shift top disk from tower 3 to tower 2

Shift top disk from tower 1 to tower 2

3 disks in tower 1 are shifted to tower 2
```