**Name:- Rina Anil Baringe.**

**Roll No.:- 11**

**Assignment No.:-4.4**

**Title:-Implementation of program based on Merge Sort Technique.**

**-------------------------------------------------------------------**#include<iostream.h>

#include<conio.h>

#include<stdlib.h>

class LIST

{

int \*A,n;

public:

LIST(int);

void READ\_LIST\_011();

void SHOW\_LIST\_011();

void MERGE\_SORT\_ASC\_011(int,int);

void MERGE\_SORT\_DESC\_011(int,int);

void MERGE\_ASC\_011(int,int,int);

void MERGE\_DESC\_011(int,int,int);

};

LIST::LIST(int par)

{

n=par;

A=new int[n+1];

};

void LIST::READ\_LIST\_011()

{

cout<<endl<<"Enter elements";

for(int i=1;i<=n;i++)

//cin>>A[i];

A[i]=random(999);

}

void LIST::SHOW\_LIST\_011()

{

cout<<endl;

for(int i=1;i<=n;i++)

cout<<A[i]<<" ";

}

void LIST::MERGE\_SORT\_ASC\_011(int low,int high)

{

if(low<high)

{

int mid=(low+high)/2;

MERGE\_SORT\_ASC\_011(low,mid);

MERGE\_SORT\_ASC\_011(mid+1,high);

MERGE\_ASC\_011(low,mid,high);

}

}

void LIST::MERGE\_SORT\_DESC\_011(int low,int high)

{

if(low<high)

{

int mid=(low+high)/2;

MERGE\_SORT\_DESC\_011(low,mid);

MERGE\_SORT\_DESC\_011(mid+1,high);

MERGE\_DESC\_011(low,mid,high);

}

}

void LIST::MERGE\_ASC\_011(int low,int mid,int high)

{

int \*B=new int[n+1];

int h=low,i=low,j=mid+1;

while(h<=mid && j<=high)

{

if(A[h] < A[j])

{

B[i]=A[h];

h=h+1;

}

else

{

B[i]=A[j];

j=j+1;

}

i=i+1;

}

if(h>mid)

{

while(j<=high)

{

B[i]=A[j];

j=j+1;

i=i+1;

}

}

else

{

while(h<=mid)

{

B[i]=A[h];

h=h+1;

i=i+1;

}

}

for(i=low;i<=high;i++)

A[i]=B[i];

}

void LIST::MERGE\_DESC\_011(int low,int mid,int high)

{

int \*B=new int[n+1];

int h=low,i=low,j=mid+1;

while(h<=mid && j<=high)

{

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

{

B[i]=A[h];

h=h+1;

}

else

{

B[i]=A[j];

j=j+1;

}

i=i+1;

}

if(h>mid)

{

while(j<=high)

{

B[i]=A[j];

j=j+1;

i=i+1;

}

}

else

{

while(h<=mid)

{

B[i]=A[h];

h=h+1;

i=i+1;

}

}

for(i=low;i<=high;i++)

A[i]=B[i];

}

void main()

{

int no;

clrscr();

cout<<endl<<"\n Enter no of element : ";

cin>>no;

LIST obj(no);

obj.READ\_LIST\_011();

cout<<endl<<"list before Sorting : ";

obj.SHOW\_LIST\_011();

obj.MERGE\_SORT\_ASC\_011(1,no);

//obj.MERGE\_SORT\_DESC\_011(1,no);

cout<<endl<<"List after Sorting : ";

obj.SHOW\_LIST\_011();

getch();

}

**Name:- Rina Anil Baringe.**

**Roll No.:-11**

**Assignment No.:-4.2(b)**

**Title:-Implementation of program based on Binary Search technique.**

**--------------------------------------------------------------------**

#include<iostream.h>

#include<conio.h>

#include<stdlib.h>

class LIST

{

int \*A,n;

public:

LIST(int);

void READ\_LIST\_011();

void SHOW\_LIST\_011();

int BINARY\_SEARCH\_011(int);

int BINARY\_SEARCH2\_011(int);

void BUBBLE\_SORT\_011();

};

LIST::LIST(int par)

{

n=par;

A=new int[n+1];

}

void LIST:: READ\_LIST\_011()

{

for(int i=1;i<=n;i++)

A[i]=random(999);

}

void LIST::SHOW\_LIST\_011()

{

for(int i=1;i<=n;i++)

cout<<A[i]<<" ";

}

void LIST::BUBBLE\_SORT\_011()

{

int flag=0;

for(int i=1;i<=n-1;i++)

{

for(int j=1 ;j<=n-i;j++)

{

if(A[j]>A[j+1])

{

flag=1;

int temp=A[j];A[j]=A[j+1];A[j+1]=temp;

}

}

if(flag==0)

break;//exit loop

}

}

int LIST::BINARY\_SEARCH\_011(int x)

{

int low=1,high=n,mid;

while(low<=high)

{

mid=(low+high)/2;

if(x==A[mid])

return mid;

else

{

if(x<A[mid])

high=mid-1;

else

low=mid+1;

}

}

return 0;

}

int LIST::BINARY\_SEARCH2\_011(int x)

{

int low=1,high=n+1,mid;

while(low<high-1)

{

mid=(low+high)/2;

if(x<A[mid])

high=mid;

else

low=mid;

}

if(x==A[low])

return low;

else

return 0;

}

void main()

{

clrscr();

int size,x,pos;

cout<<"Enter size of list: ";

cin>>size;

LIST obj(size);

obj.READ\_LIST\_011();

obj.SHOW\_LIST\_011();

obj.BUBBLE\_SORT\_011();

cout<<endl<<"After sorting list: ";

obj.SHOW\_LIST\_011();

do

{

cout<<endl<<"Enter an element to found"<<endl;

cin>>x;

pos=obj.BINARY\_SEARCH\_011(x);

//pos=obj.BINARY\_SEARCH2\_011(x);

if(pos>0)

cout<<endl<<"Element is found at "<<pos<<endl;

else

cout<<endl<<"Element is not found";

}while(1);

}

**Name:- Rina Anil Baringe.**

**Roll No.:- 11**

**Assignment No.:- 4.6**

**Title:-Implementation of program based on searching technique-Radix Sort.**

#include<iostream.h>

#include<conio.h>

#include<stdlib.h>

class LIST

{

int\*A,n;

public:

LIST(int);

void READ\_LIST\_011();

void SHOW\_LIST\_011();

void RADIX\_SORT\_ASC\_011();

void RADIX\_SORT\_DESC\_011();

};

LIST::LIST(int par)

{

n=par;

A=new int[n+1];

}

void LIST::READ\_LIST\_011()

{

for(int i=1;i<=n;i++)

{

A[i]=random(32000);

}

}

void LIST::SHOW\_LIST\_011()

{

cout<<endl;

for(int i=1;i<=n;i++)

cout<<" "<<A[i];

}

void LIST::RADIX\_SORT\_ASC\_011()

{

//create queue and front,rear pointer for 10 queues

int \*Q[10];

for(int p=0;p<10;p++)

Q[p]=new int[n+1];

int F[10];

int R[10];

/////////////////////////////////////////////////

int k=1;

for(int d=1;d<=5;d++)

{

for(int j=0;j<10;j++)

{

F[j]=0;R[j]=0;

}

///////////////////////////////////////////

for(int i=1;i<=n;i++)

{

int qi=(A[i]/k)%10;

if(F[qi]==0)

F[qi]=1;

R[qi]=R[qi]+1;

Q[qi][R[qi]]=A[i];

}

cout<<endl;

for(i=1,j=0;j<10;j++)

{

for(int m=1;m<=R[j];m++)

{

A[i]=Q[j][m];

i=i+1;

}

}k=k\*10;

}

}

void LIST::RADIX\_SORT\_DESC\_011()

{

int \*Q[10];

for(int p=0;p<10;p++)

Q[p]=new int[n+1];

int F[10];

int R[10];

int k=1;

for(int d=1;d<=5;d++)

{

for(int j=0;j<=10;j++)

{

F[j]=0;R[j]=0;

}

for(int i=1;i<=n;i++)

{

int qi=(A[i]/k)%10;

if(F[qi]==0)

F[qi]=1;

R[qi]=R[qi]+1;

Q[qi][R[qi]]=A[i];

}

cout<<endl;

for(i=1,j=9;j>=0;j--)

{

for(int m=1;m<=R[j];m++)

{

A[i]=Q[j][m];

i=i+1;

}

}k=k\*10;

}

}

void main()

{

int size;

clrscr();

cout<<endl<<"Enter size of List: ";

cin>>size;

LIST obj(size);

obj.READ\_LIST\_011();

cout<<endl<<"List Element Before Sorting: \n";

obj.SHOW\_LIST\_011();

obj.RADIX\_SORT\_ASC\_011();

//obj.RADIX\_SORT\_DESC\_011();

cout<<endl<<"List element After sorting: \n";

obj.SHOW\_LIST\_011();

getch();

}

**Name:- Rina Anil Baringe.**

**Roll No.:- 11**

**AssignmentNo.:-4.2(a)**

**Title:-Implementation of program based on Linear Search Technique.**

**--------------------------------------------------------------------**#include<iostream.h>

#include<conio.h>

#include<stdlib.h>

class LIST

{

int \*A,n;

public:

LIST(int);

void READ\_LIST\_011();

void SHOW\_LIST\_011();

int LINEAR\_SEARCH\_011(int);

void BUBBLE\_SORT\_011();

};

LIST::LIST(int par)

{

n=par;

A=new int[n+1];

}

void LIST:: READ\_LIST\_011()

{

for(int i=1;i<=n;i++)

A[i]=random(999);

}

void LIST::SHOW\_LIST\_011()

{

for(int i=1;i<=n;i++)

cout<<A[i]<<" ";

}

void LIST::BUBBLE\_SORT\_011()

{

int flag=0;

for(int i=1;i<=n-1;i++)

{

for(int j=1 ;j<=n-i;j++)

{

if(A[j]>A[j+1])

{

flag=1;

int temp=A[j];A[j]=A[j+1];A[j+1]=temp;

}

}

if(flag==0)

break;//exit loop

}

}

int LIST::LINEAR\_SEARCH\_011(int x)

{

for(int i=1;i<=n;i++)

{

if(A[i]==x)

{

return i;

}

}

return 0;

}

void main()

{

clrscr();

int size,x,pos;

cout<<"Enter size of list: ";

cin>>size;

LIST obj(size);

obj.READ\_LIST\_011();

obj.SHOW\_LIST\_011();

obj.BUBBLE\_SORT\_011();

cout<<endl<<"After sorting list:"<<endl;

obj.SHOW\_LIST\_011();

do

{

cout<<endl<<"Enter an element to found"<<endl;

cin>>x;

pos=obj.LINEAR\_SEARCH\_011(x);

if(pos>0)

cout<<endl<<"Element is found at "<<pos<<endl;

else

cout<<endl<<"Element is not found";

}while(1);

}

**Name: Rina Anil Baringe.**

**Roll No: 11**

**Assignment No.:- 4.5**

**Title:Implementation of program based on Quick Sort technique.**

**-------------------------------------------------------------------------------------------------**

#include<iostream.h>

#include<conio.h>

#include<stdlib.h>

class LIST

{

int \*A,n;

public:

LIST(int);

void READ\_LIST\_011();

void SHOW\_LIST\_011();

void QUICK\_SORT\_ASC\_011(int,int);

void QUICK\_SORT\_DESC\_011(int,int);

int PARTITION\_ASC\_011(int,int);

int PARTITION\_DESC\_011(int,int);

};

LIST::LIST(int par)

{

n=par;

A=new int[n+2];

}

void LIST::READ\_LIST\_011()

{

cout<<endl<<"Enter Elements:";

for(int i=1;i<=n;i++)

{

//cin>>A[i];

A[i]=random(999);

}

A[i]=9999;//positive infinity

}

void LIST::SHOW\_LIST\_011()

{

cout<<endl;

for(int i=1;i<=n;i++)

{

cout<<A[i]<<" ";

}

}

void LIST::QUICK\_SORT\_ASC\_011(int p,int q)

{

if(p<q)

{

int j=q+1;

j=PARTITION\_ASC\_011(p,j);

QUICK\_SORT\_ASC\_011(p,j-1);

QUICK\_SORT\_DESC\_011(j+1,q);

}

}

void LIST::QUICK\_SORT\_DESC\_011(int p,int q)

{

if(p<q)

{

int j=q+1;

j=PARTITION\_DESC\_011(p,j);

QUICK\_SORT\_ASC\_011(p,j-1);

QUICK\_SORT\_DESC\_011(j+1,q);

}

}

int LIST::PARTITION\_ASC\_011(int m,int p)

{

int temp=A[m];

int i=m;

do

{

do

{

i=i+1;

}

while(A[i]<temp);

do

{

p=p-1;

}

while(A[p]>temp);

if(i<p)

{

int t=A[i];A[i]=A[p];A[p]=t;

}

else

break;

}

while(1);

A[m]=A[p];

A[p]=temp;

return p;

}

int LIST::PARTITION\_DESC\_011(int m,int p)

{

int temp=A[m];

int i=m;

do

{

do

{

i=i+1;

}

while(A[i]>temp);

do

{

p=p-1;

}

while(A[p]<temp);

if(i<p)

{

int t=A[i];A[i]=A[p];A[p]=t;

}

else

break;

}

while(1);

A[m]=A[p];

A[p]=temp;

return p;

}

void main()

{

int no;

clrscr();

cout<<endl<<"\n Enter no of Elements:";

cin>>no;

LIST obj(no);

obj.READ\_LIST\_011();

cout<<endl<<"List before Sorting:";

obj.SHOW\_LIST\_011();

obj.QUICK\_SORT\_ASC\_011(1,no);

// obj.QUICK\_SORT\_DESC\_011(1,no);

cout<<endl<<"List after Sorting:";

obj.SHOW\_LIST\_011();

getch();

}