/\*

***Name* :-** Abrar Mahedavi

***Roll No.* :-** 21035

***Title* :-** Write a modular program using object oriented programming features to implement different sorting methods like quick, merge, shell, radix .sort using c++.

/\*

***Program :***

#include<iostream>

using namespace std;

class sort //Declaration of Class

{

private:

int array[100],c[100];

public:

void input(int n); //Prototype for input function

void old(int); //Prototype for last Input data

void display(int n); //Prototype for Display function

int qsort(int,int); //Prototype for Quick Sort function

void Partition(int,int); //Prototype for Partition function

void sort1(int low,int mid,int high); //Prototype for Merging array function

void divide(int low,int high); //Prototype for Dividing array function

void shellsort(int); //Prototype for Shell Sort function

void radixsort(int); //Prototype for radix sorting function

};

void sort::input(int n) //Function Definition for Input

{

int i;

cout<<"Enter the Elements in array \t"<<endl;

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

{

cout<<"\t";

cin>>array[i];

c[i]=array[i];

}

}

void sort::old(int n) //Function Definition for previous Input

{

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

array[i]=c[i];

}

int sort :: qsort(int o,int n) //Function Definition for Quick Sort

{

int up,down, pivot,temp,i;

pivot=array[o]; up=o; down=n;

while(down>up)

{

while(array[up]<=pivot)

{

up++;

}

while(array[down]>pivot)

{

down--;

}

if(up<down) //Swapping of elements in array

{

temp= array[up];

array[up]=array[down];

array[down]=temp;

}

} //Swapping of pivot Element

array[o]=array[down];

array[down]=pivot;

return down;

}

void sort :: Partition(int up,int down) //Function Definition for Partition of

{ //array

int pivot;

if(down>up)

{

pivot=qsort(up,down);

Partition(up,pivot-1);

Partition(pivot+1,down);

}

}

void sort::divide(int low,int high) //Function Definition for Dividing

{ //array

if(low<high)

{

int mid;

mid=(low+high)/2;

divide(low,mid);

divide(mid+1,high);

sort1(low,mid,high); //Calling of Sort and Merge function

}

}

void sort::sort1(int low,int mid,int high) //Function Definition for Sorting merge {

int i,j,k;

i=low;

j=mid+1;

k=low;

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

{

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

{

c[k]=array[i];

k++;

i++;

}

else

{

c[k]=array[j];

k++;

j++;

}

}

while(i<=mid)

{

c[k]=array[i];

k++;

i++;

}

while(j<=high)

{

c[k]=array[j];

k++;

j++;

}

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

{

array[i]=c[i];

}

}

void sort::shellsort(int n) //Function Definition for Shell Sort

{

int temp,gap,i;

int swapped;

gap=n/2;

do

{

do

{

swapped=0;

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

{

if(array[i]>array[i+gap]) //Checking of condition whether Element //is smaller or greater

{

temp=array[i];

array[i]=array[i+gap];

array[i+gap]=temp;

swapped=1;

}

}

}while(swapped==1); //To Swap the elements

}while(gap=(gap/2)>=1); //To increase the gap

}

void sort::radixsort(int n) //Function Definition for Radix Sort

{

int largest,divisor=1,r=0;

int bucket[10][15],counter[10];

largest=array[0];

for(int i=1;i<n;i++) //For Checking Largest Number in array

{

if(array[i]>largest)

{

largest=array[i];

}

}

cout<<"Largest number in a is"<<largest<<endl;

int p=0;

while(largest>0) // Printing the number of passes in array

{

p++;

largest=largest/10;

}

cout<<"Total number of pass are "<<p<<endl;

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

{

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

{

counter[q]=0;

}

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

{

r=(array[j]/divisor)%10;

bucket[r][counter[r]]=array[j];

counter[r]++;

}

int z=0;

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

{

for(int b=0;b<counter[w];b++)

{

array[z++]=bucket[w][b];

}

} divisor=divisor\*10;

}

}

void sort::display(int n) //Function Definition for Display

{

int i;

cout<<"Sorted array is "<<endl;

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

{

cout<<"\t"<<array[i]<<endl;

}

}

int main()

{

sort s;

int n,num;

char c,c1;

cout<<"\nEnter the number of Elements of the array \t";

cin>>n;

cout<<"Total number of Elements are "<<n<<endl;

if(n==0)

{

cout<<"Entered number is 0 alreadry sorted"<<endl;

}

else

{

if(n==1)

{

s.input(n);

cout<<"Entered number is 1 already sorted"<<endl;

s.display(n);

}

else

{

s.input(n);

}

do

{

cout<<"\*\*\*\*\*\*\*\*\*\*Differnt type of sort\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"\t1 Quick Sort"<<endl;

cout<<"\t2 Merge Sort"<<endl;

cout<<"\t3 Shell Sort"<<endl;

cout<<"\t4 Radix Sort"<<endl;

cout<<"Enter type of sort u want to perform \t";

cin>>num;

cout<<"Enter Choice is "<<num<<endl;

switch(num)

{

case 1:cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*QuickSort\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl<<endl;

s.Partition(0,n-1);

//Calling of Partition(in which Quick sort function is called ) Function

s.display(n);

break;

case 2:cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*Merge Sort\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl<<endl;

s.divide(0,n-1);

//Calling of Dividing (in which sort and merge function is called) Function

s.display(n);

break;

case 3:cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*Shell Sort\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl<<endl;

s.shellsort(n); //Calling of Shell Sort Function

s.display(n);

break;

case 4:cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*Radix Sort\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl<<endl;

s.radixsort(n); //Calling of Radix Sort Function

s.display(n);

break;

default:cout<<"Entered Choice is Wrong"<<endl<<endl;

break;

}

cout<<"Do u want to continue Yes or No \t"<<endl;

cin>>c;

if(c=='y' || c=='Y')

{

cout<<"Do u want to continue With same numbers yes or no \t"<<endl;

cin>>c1;

if(c1=='y' || c1=='Y')

{

s.old(n);

}

else

{

cout<<"Enter the number of Elements of the array \t";

cin>>n;

cout<<"Total number of Elements are "<<n<<endl;

s.input(n);

}

}

}while(c=='y' || c=='Y');

}

return 0;

}

***Output :***

compeng-sl2-08@compeng-sl2-08:~/Abrar$ g++ Assignment1.cpp

compeng-sl2-08@compeng-sl2-08:~/Abrar$ ./a.out

Enter the number of Elements of the array 0

Total number of Elements are 0

Entered number is 0 already sorted .

Do u want to continue Yes or No

y

Do u want to continue With same numbers yes or no

n

Enter the number of Elements of the array 1

Total number of Elements are 1

Enter the Elements in array

6

Entered number is 1 already sorted.

Do u want to continue Yes or No

y

Do u want to continue With same numbers yes or no

n

Enter the number of Elements of the array 9

Total number of Elements are 9

Enter the Elements in array

20

45

16

65

105

16

8

0

98

\*\*\*\*\*\*\*\*\*\*Differnt type of sort\*\*\*\*\*\*\*\*\*\*

1 Quick Sort

2 Merge Sort

3 Shell Sort

4 Radix Sort

Enter type of sort u want to perform 1

Enter Choice is 1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*QuickSort\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sorted array is

0

8

16

16

20

45

65

98

105

Do u want to continue Yes or No

y

Do u want to continue With same numbers yes or no

Y

\*\*\*\*\*\*\*\*\*\*Differnt type of sort\*\*\*\*\*\*\*\*\*\*

1 Quick Sort

2 Merge Sort

3 Shell Sort

4 Radix Sort

Enter type of sort u want to perform 2

Enter Choice is 2

\*\*\*\*\*\*\*\*\*\*\*\*\*\*Merge Sort\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sorted array is

0

8

16

16

20

45

65

98

105

Do u want to continue Yes or No

y

Do u want to continue With same numbers yes or no

n

Enter the number of Elements of the array 5

Total number of Elements are 5

Enter the Elements in array

-80

-1

120

10

5

\*\*\*\*\*\*\*\*\*\*Differnt type of sort\*\*\*\*\*\*\*\*\*\*

1 Quick Sort

2 Merge Sort

3 Shell Sort

4 Radix Sort

Enter type of sort u want to perform 3

Enter Choice is 3

\*\*\*\*\*\*\*\*\*\*\*\*\*\*Shell Sort\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sorted array is

-80

-1

5

10

120

Do u want to continue Yes or No

Y

Do u want to continue With same numbers yes or no

N

Enter the number of Elements of the array 10

Total number of Elements are 10

Enter the Elements in array

10

14

19

8523

457

325

985

21

5

265

\*\*\*\*\*\*\*\*\*\*Differnt type of sort\*\*\*\*\*\*\*\*\*\*

1 Quick Sort

2 Merge Sort

3 Shell Sort

4 Radix Sort

Enter type of sort u want to perform 4

Enter Choice is 4

\*\*\*\*\*\*\*\*\*\*\*\*\*\*Radix Sort\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Largest number in a is8523

Total number of pass are 4

Sorted array is

5

10

14

19

21

265

325

457

985

8523

Do u want to continue Yes or No

N