## Implement Bubble Sort and Selection Sort

# include <iostream> # include <conio.h> using namespace std; class sorting

{

int a [10],n; public :

void getdata(); void putdata(); void bubblelogic();

void selectionlogic();

};

void sorting :: getdata()

{

cout<<"Enter the range of the array: "; cin >> n;

cout << "\nEnter the elements of the array : "; for( int i = 0; i<n; i++)

{

cin >> a[i];

}

}

void sorting :: putdata()

{

cout<< "\nAfter sorting array is : "; for (int i =0 ; i<n; i++)

{

cout <<a[i] << "\t";

}

}

void sorting ::

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

{

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

{

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

{

int temp = a[i]; a[i] = a[j];

a[j] = temp;

}

}

}

}

void sorting :: selectionlogic(){

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

{

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

{

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

{

int temp = a[i]; a[i] = a[j];

a[j] = temp;

}

}

}

}

int main()

{

sorting s; int c; while (1)

{

cout << "\n1. Bubble sort:"; cout << "\n2. Selection sort:"; cout << "\n3. Exit:";

cout << "\nEnter your choice:"; cin >> c;

switch (c)

{

case 1: s.getdata(); s.bubblelogic(); s.putdata(); break;

case 2: s.getdata(); s.selectionlogic(); s.putdata(); break;

case 3: exit(1);

}

}

getch();

}

# To Implement Insertion Sort & Shell Sort

## Insertion Sort

# include <iostream> # include <conio.h> using namespace std;

class sorting

{

int a [10],n; public :

};

void getdata(); void putdata(); void insertion();

void sorting :: getdata()

{

//definig the size of the array

cout<<"Enter the range of the array: "; cin >> n;

// defining the elements of the array

cout << "\nEnter the elements of the array : "; for( int i = 0; i<n; i++)

{

cin >> a[i];

}

}

void sorting :: putdata()

{

cout<< "\n\n Sorted array is : "; for (int i =0 ; i<n; i++)

{

cout <<a[i];

}

}

void sorting :: insertion(){

for (int i = 1; i < n; i++) // Start from the second element

{

int temp = a[i]; int j = i - 1;

while (j >= 0 && a[j] > temp)

{

a[j + 1] = a[j];

j = j - 1;

}

a[j + 1] = temp;

}

}

int main()

{

sorting s; s.getdata(); s.insertion(); s.putdata();

getch();

}

## Shell Sort

# include <iostream> # include <conio.h> using namespace std;

class sort

{

};

private: public:

int arr[10],n; void get();

void shellsort(); void put();

void sort :: get()

{

cout<<"Enter the size of the array : "; cin>>n;

cout<<"Enter the elements of the Array : "; for(int i=0; i<n; i++)

{

cin>>arr[i];

}

}

void sort::shellsort()

{

for(int incr = n/2; incr>0; incr/=2)

{

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

{

int temp = arr[i]; int j;

for(j=i; j>=incr&&arr[j-incr]>temp;j-=incr)

{

arr[j] = arr[j-incr];

}

arr[j] =temp;

}

}

}

void sort :: put()

{

cout<<"\nThe Array is sorted using the shell sort : \n"; for(int i=0; i<n; i++)

{

}

int main()

{

}

sort s; s.get();

cout<<arr[i]<<" ";

s.shellsort(); s.put();

getch();

# To Implement Radix sort

## Radix Sort

# include <iostream> # include <conio.h> using namespace std;

class radix

{

int A[20],n,i; public:

void getdata();

void radsort(int\*,int); void display(int\*,int);

};

void radix::getdata()

{

cout<<"Enter the size of array: "; cin>>n;

cout<<"Enter the elements to sort: "; for(i=0;i<n;i++)

{

cin>>A[i];

}

radsort(A,n);

display(A,n);

}

void radix::radsort(int \*A,int n)

{

int temp;

int bucket[10][20]; int buck\_count[10];

int i,k,j,r,no\_of\_passes=0,divisor=1; int largest,pass\_no;

largest=A[0]; for(i=1;i<n;i++)

{

if(A[i]>largest) largest=A[i];

}

while(largest>0)

{

no\_of\_passes++; largest=largest/10;

}

for(pass\_no=0;pass\_no<no\_of\_passes;pass\_no++)

{

for(k=0;k<10;k++) buck\_count[k]=0; for(i=0;i<n;i++)

{

} i=0;

r=(A[i]/divisor)%10; bucket[r][buck\_count[r]++]=A[i];

for(k=0;k<10;k++)

{

for(j=0;j<buck\_count[k];j++) A[i++]=bucket[k][j];

}

divisor=divisor\*10;

}

}

void radix::display(int\* A,int n)

{

int i;

cout<<"\n\t Sorted array:\n\t"; for(i=0;i<n;i++) cout<<A[i]<<"\n\t";

}

int main()

{

radix r;

r.getdata();

getch();

}