**SUBMITTED BY**

**LALIT AGGARWAL (067)**

**PRATEEK TANEJA (065)**

**CSE, BVCOE**

**SOURCE CODE**

/\*................HEADER FILES.............\*/

#include<graphics.h>

#include<stdio.h>

#include<conio.h>

#include<dos.h>

#include<process.h>

/\*................FUNCTION PROTOTYPES.............\*/

void draw();

void menu();

void object(int,int,int);

void bubble(int a[],int);

void shell(int a[],int);

void selection\_sort(int a[],int);

void quick\_sort(int a[],int,int);

void move\_vertical(int a[],int,int);

void move\_horizontal(int a[],int,int);

int size,gm=0,gd=0;

/\*................MAIN FUNCTION.............\*/

void main()

{

clrscr();

initgraph(&gd,&gm,"C:\\TC\\BGI");

menu();

closegraph();

getch();

}

/\*................MENU FUNCTION.............\*/

void menu()

{

int i,size,a[10],p,r;

char ch,choice;

do

{

cleardevice();

printf("Enter the No. of Elements : ");

scanf("%d",&size);

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

{

printf("\nEnter Element %d: ",i+1);

scanf("%d",&a[i]);

}

cleardevice();

settextstyle(3,0,2);

outtextxy(0,0,"A : Bubble Sort");

outtextxy(0,20,"B : Shell Sort");

outtextxy(0,40,"C : Selection Sort");

outtextxy(0,60,"D : Quick Sort");

outtextxy(0,80,"E : Exit");

outtextxy(0,100,"Press your choice : ");

ch=getch();

switch(ch)

{

case 'a':bubble(a,size);

break;

case 'b':shell(a,size);

break;

case 'c':selection\_sort(a,size);

break;

case 'd':p=0;

r=size-1;

cleardevice();

draw();

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

{

object(100+i\*50,150,a[i]);

getch();

}

quick\_sort(a,p,r);

draw();

getch();

cleardevice();

break;

case 'e':exit(0);

default :cleardevice();

outtextxy(0,100,"WRONG CHOICE!!!!!!");

}

fflush(stdin);

outtextxy(0,0,"Do you want to continue(y/n) : ");

choice=getch();

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

}

/\*................BUBBLE SORT.............\*/

/\*................DECREASING ORDER.............\*/

void bubble(int a[],int n)

{

int i,j,temp;

cleardevice();

draw();

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

{

object(100+i\*50,150,a[i]);

getch();

}

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

for(j=0;j<n-1-i;j++)

if(a[j]<a[j+1])

{

move\_vertical(a,j,j+1);

temp=a[j];

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

a[j+1]=temp;

}

draw();

getch();

cleardevice();

}

/\*................SHELL SORT.............\*/

/\*................INCREASING ORDER.............\*/

void shell(int a[],int n)

{

int i,j,k,m,mid;

cleardevice();

draw();

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

{

object(100+i\*50,150,a[i]);

getch();

}

for(m=n/2;m>0;m/=2)

{

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

{

for(i=j-m;i>=0;i-=m)

{

if(a[i]>a[i+m])

{

move\_vertical(a,i,i+m);

mid = a[i];

a[i] = a[i+m];

a[i+m] = mid;

}

else

break;

}

}

}

draw();

getch();

cleardevice();

}

/\*................SELECTION SORT.............\*/

/\*................DECREASING ORDER.............\*/

void selection\_sort(int a[],int n)

{

int i,j,temp,min,loc;

cleardevice();

draw();

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

{

object(100+i\*50,150,a[i]);

getch();

}

for(i=0;i<n-1;i++)

{

min=a[i];

loc=i;

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

{

if(a[j]<min)

{

min=a[j];

loc=j;

}

}

if(i!=loc)

{

move\_vertical(a,i,loc);

temp=a[i];

a[i]=a[loc];

a[loc]=temp;

}

}

draw();

getch();

cleardevice();

}

/\*................QUICK SORT.............\*/

/\*................INCREASING ORDER.............\*/

int partition(int a[],int p,int r)

{

int j,x=a[r],i=p-1,temp;

for(j=p;j<=r-1;j++)

{

if(a[j]<=x)

{

i=i+1;

move\_vertical(a,i,j);

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

move\_vertical(a,i+1,r);

temp=a[i+1];

a[i+1]=a[r];

a[r]=temp;

return i+1;

}

void quick\_sort(int a[],int p,int r)

{

int q,i;

if(p<r)

{

q=partition(a,p,r);

quick\_sort(a,p,q-1);

quick\_sort(a,q+1,r);

}

}

/\*................DRAW FUNCTION.............\*/

void draw()

{

settextstyle(3,0,2);

outtextxy(250,50,"GIVEN NUMBERS ");

outtextxy(250,260,"AFTER SORTING ");

line(0,getmaxy()/2,getmaxx(),getmaxy()/2);

}

/\*................OBJECT FUNCTION.............\*/

void object(int x,int y,int no)

{

char s[8];

sprintf(s,"%d",no);

circle(x,y,15);

settextstyle(2,0,6);

outtextxy(x-3,y-10,s);

}

/\*................MOVING VERTICAL FUNCTION.............\*/

void move\_vertical(int a[],int f,int s)

{

int i;

for(i=0;i<50;i++)

{

setcolor(WHITE);

object(100+f\*50,150+i\*4,a[f]);

object(100+s\*50,150+i\*4,a[s]);

delay(15);

setcolor(BLACK);

object(100+f\*50,150+i\*4,a[f]);

object(100+s\*50,150+i\*4,a[s]);

}

setcolor(WHITE);

object(100+f\*50,150+i\*4,a[f]);

object(100+s\*50,150+i\*4,a[s]);

move\_horizontal(a,f,s);

for(i=50;i>0;i--)

{

setcolor(WHITE);

object(100+f\*50,150+i\*4,a[s]);

object(100+s\*50,150+i\*4,a[f]);

delay(15);

setcolor(0);

object(100+f\*50,150+i\*4,a[s]);

object(100+s\*50,150+i\*4,a[f]);

}

setcolor(WHITE);

object(100+f\*50,150+i\*4,a[s]);

object(100+s\*50,150+i\*4,a[f]);

}

/\*................MOVING HORIZONTAL FUNCTION.............\*/

void move\_horizontal(int a[],int f,int s)

{

int i;

for(i=0;i<(s-f)\*50;i++)

{

setcolor(WHITE);

object(100+f\*50+i,350,a[f]);

object(100+s\*50-i,350,a[s]);

delay(20);

setcolor(0);

object(100+f\*50+i,350,a[f]);

object(100+s\*50-i,350,a[s]);

}

setcolor(WHITE);

object(100+f\*50+i,350,a[f]);

object(100+s\*50-i,350,a[s]);

}