package lg\_Project;

import java.util.Scanner;

class arrayfactory{ // start of the class array factory

Scanner scin = new Scanner(System.in);

int arrayth;

arrayfactory(){}

arrayfactory(int arrri)// constructor

{

arrayth = arrri;

}

int[] makearray() { //method which makes the array

int[] kill = new int[arrayth];

for(int a =0;a<kill.length;a++) {

System.out.println("type in the values");

kill[a] = scin.nextInt();

}

return kill;// returns the array kill

}// end of the method makearray

}// end of the class arrayfactory

public class ALLINONEALOGORITHM {static // another class allinonealgorithm

void bubbleSort(int[] sort) {// bubble sort algorithm, continuously sorts the array by swapping the value with the next value if the previous value is bigger than the next one.

int n = sort.length;

int temp = 0;

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

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

if(sort[j-1] > sort[j]) { // swaps the values if the index j-1 is bigger than the value j

temp = sort[j-1];

sort[j-1] = sort[j];

sort[j] = temp;

}

}

}

}

static void linearsearch(int[] sort,int p) {// linear search algorithm

int y = sort.length;

for(int a=0;a<y;a++){

if(sort[a]==p) {//simple algorithm which searches the index one by one until it finds our the value

String term2;

if(a==1) {

term2 ="st";

}

else if(a==2) {

term2="nd";

}

else if(a==3) {

term2="rd";

}

else {

term2="th";

}

System.out.println("the element is located in "+a+term2+" index of the array");

}

}

}

static void binarysearch(int[] sort,int u) { // method for the binary search

int k=0;

int j=sort.length-1;

while(k<=j) {

int mid = (j+k)/2;// finds the mid point

if(sort[mid]<u) {

k= mid+1;// if the mid point value is smaller than the value which we want to find, new minimum = mid index + 1

}

else if(sort[mid]>u) {

j= mid-1;//if the mid point value is larger than the value which we want to find, new minimum = mid index -1

}

else

{

String term;

if(mid==1) {

term ="st";

}

else if(mid==2) {

term="nd";

}

else if(mid==3) {

term="rd";

}

else {

term="th";

}

System.out.println("the element is located in "+mid+term+" index of the array");

k=j+1;

}

}

}

public static int[] hou(int sort[]) { // Selection sort algorithm

int k = sort.length;

for(int j =0;j<k-1;j++) {// searches through an unsorted array

int min\_idx = j;

for(int u= j+1;u<k;u++) {

if(sort[min\_idx]>sort[u]) {

int temp = sort[min\_idx]; //swaps the minimum value in the unsorted array with the first value of the minimum index

sort[min\_idx] = sort[u];

sort[u] = temp;

}

}

}

return sort;

}// end of the selection sort algorithm

static void printArray(int sort[]) // method which would print the array

{

int n = sort.length;

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

System.out.print(sort[i]+" ");

System.out.println();

} // end of the method

public static void main(String[] args) {// main method

// TODO Auto-generated method stub

Scanner v = new Scanner(System.in);

System.out.println("type in the length of the array");

int ggg = v.nextInt();

arrayfactory plz = new arrayfactory(ggg);// creating a new instance in the class arrayfactory

int[] sort = plz.makearray();

System.out.println("\n\nwould you like to print an unsorted array?\n\nY or N\n\n");

String YORN = v.next();

boolean kerro = false;// just meaningless name,

while(kerro==false) {// this algorithm finds out

if (YORN.equals("Y")) {

System.out.println("you've selected yes\n\n");

printArray(sort); // calls an unsorted array

System.out.println("\n\n");

kerro=true;// sets it true so that it doesn't iterates once again

}

else if (YORN.equals("N")) {

System.out.println("\n\nyou've selected no\n\n");

kerro=true;// sets it true so that it doesn't iterates once again

}

else {

System.out.println("your choice is invalid try it again(Try it in capital letter)\n\nwould you like to print an unsorted array?\n\nY or N\n\n");

YORN = v.next();

}

}

System.out.println("pick an algorithm which you would prefer to use in sorting the array\n 1:bubble sort\n 2:selection sort");

int ooo = v.nextInt();// lets the user pick between which sort to use

boolean kimchi = false;

while(kimchi==false) {

if(ooo==1)

{

System.out.println("you've selected bubble sort\n\n\n\n");

bubbleSort(sort);// calls the bubble sort

kimchi=true;// kicks it out of the loop

}

else if(ooo==2)

{

System.out.println("you've selected selection sort\n\n\n\n");

hou(sort);// calls the selection sort

kimchi=true;// kicks it out of the loop

}

else

{

System.out.println("your choice is not valid try it again\n\npick an algorithm which you would prefer to use in sorting the array\n 1:bubble sort\n 2:selection sort");

ooo=v.nextInt(); //lets the user choose the number again if the user chooses an invalid number

}

}

System.out.println("pick an algorithm which you would prefer to use in searching the element\n 1:binary search\n 2:linear search\n");

int uuu = v.nextInt();// lets the user choose which searching algorithm to use

boolean cmon = false;

while(cmon==false) {

if(uuu==1)

{

System.out.println("please type in the element which you would like to know where it is located in the array");

int kkk = v.nextInt();

binarysearch(sort,kkk);// calls the

cmon=true;

}

else if(uuu==2)

{

System.out.println("please type in the element which you would like to know where it is located in the array");

int fff = v.nextInt();

linearsearch(sort,fff);

cmon=true;

}

else

{

System.out.println("your choice is not valid try it again\n\npick an algorithm which you would prefer to use in searching the element\n 1:binary search\n 2:linear search\n");

uuu=v.nextInt();

}

}

System.out.println("\n\nwould you like to see the sorted array?\n\nY or N\n\n");

String end = v.next();// asks if the user wants to see the sorted array

boolean kim = false;

while(kim==false) {

if (end.equals("Y")) {

System.out.println("you've selected yes\n\n");

printArray(hou(sort)); // prints out the sorted array using Selection sort

System.out.println("\n\n");

kim=true;// kicks it out the loop

}

else if (end.equals("N")) {

System.out.println("\n\nyou've selected no\n\n");

kim=true;// kicks it out of the loop

}

else {

System.out.println("your choice is invalid try it again(Try it in capital letter)\n\nwould you like to print an unsorted array?\n\nY or N\n\n");

end = v.next();// lets the user choose it again if the letter chosen is invalid

}

}

}

}