**Linear Search code example:**

public class LinearSearchExample{

public static int linearSearch(int[] arr, int key){

for(int i=0;i<arr.length;i++){

if(arr[i] == key){

return i;

}

}

return -1;

}

public static void main(String a[]){

int[] a1= {10,20,30,50,70,90};

int key = 50;

System.out.println(key+" is found at index: "+linearSearch(a1, key));

}

}

**Binary Search code example:**

class BinarySearchExample{

public static void binarySearch(int arr[], int first, int last, int key){

int mid = (first + last)/2;

while( first <= last ){

if ( arr[mid] < key ){

first = mid + 1;

}else if ( arr[mid] == key ){

System.out.println("Element is found at index: " + mid);

break;

}else{

last = mid - 1;

}

mid = (first + last)/2;

}

if ( first > last ){

System.out.println("Element is not found!");

}

}

public static void main(String args[]){

int arr[] = {10,20,30,40,50};

int key = 30;

int last=arr.length-1;

binarySearch(arr,0,last,key);

}

}

**Bubble Sort code example:**

// Optimized Bubble sort in Java

import java.util.Arrays;

class Main {

// perform the bubble sort

static void bubbleSort(int array[]) {

int size = array.length;

// loop to access each array element

for (int i = 0; i < (size-1); i++) {

// check if swapping occurs

boolean swapped = false;

// loop to compare adjacent elements

for (int j = 0; j < (size-i-1); j++) {

// compare two array elements

// change > to < to sort in descending order

if (array[j] > array[j + 1]) {

// swapping occurs if elements

// are not in the intended order

int temp = array[j];

array[j] = array[j + 1];

array[j + 1] = temp;

swapped = true;

}

}

// no swapping means the array is already sorted

// so no need for further comparison

if (!swapped)

break;

}

}

}

**Driver:**

public static void main(String args[]) {

int[] data = { -2, 45, 0, 11, -9 };

// call method using the class name

Main.bubbleSort(data);

System.out.println("Sorted Array in Ascending Order:");

System.out.println(Arrays.toString(data));

}

}

**Quick Sort code example:**

void quicksort(int number[25],int first,int last){

int i, j, pivot, temp;

if(first<last){

pivot=first;

i=first;

j=last;

while(i<j){

while(number[i]<=number[pivot]&&i<last)

i++;

while(number[j]>number[pivot])

j--;

if(i<j){

temp=number[i];

number[i]=number[j];

number[j]=temp;

}

}

temp=number[pivot];

number[pivot]=number[j];

number[j]=temp;

quicksort(number,first,j-1);

quicksort(number,j+1,last);

}

}

**Driver**

int main(){

int i, count, number[25];

printf("How many elements are u going to enter?: ");

scanf("%d",&count);

printf("Enter %d elements: ", count);

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

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

quicksort(number,0,count-1);

printf("Order of Sorted elements: ");

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

printf(" %d",number[i]);

return 0;

}