**Question 01(a):**

import java.util.\*;

public class Main{

static void l\_search(int arr[], int n){

int max\_c=0;

int index=0;

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

int c=0;

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

if(arr[i]==arr[j])

c++; }

if(c>max\_c){

max\_c=c;

index=i;

break; } }

if(max\_c>n/2)

System.out.println("max element is = "+arr[index]);

else

System.out.println("not found");

}

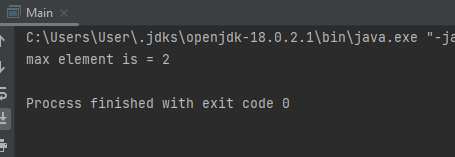
public static void main(String[] args) {

int arr[] = { 2, 2, 1, 2, 3, 5, 2 };

int n = arr.length;

l\_search(arr, n);

} }



**Question 01(b):**

import java.util.Scanner;

public class Question\_number1B {

public static void main(String[] args) {

Question\_number1B q = new Question\_number1B();

int arr[] = {13,11,9,7,5,3,1};

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

for (int j = i+1; j < arr.length ; j++) {

if (arr[i]>arr[j]){

int temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

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

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

}

Scanner sc = new Scanner(System.in);

System.out.println("\nEnter number you wanna search?");

int prob = q.find(arr,sc.nextInt());

if (prob==1){

System.out.println("Number exist");

}else {

System.out.println("Number does not exist");

}

}

public int find(int[] arr,int key){

int s = 0; int e = arr.length-1;

while (s<=e && arr[s]<key && arr[e]>key) {

int position = s + (((e - s)\*(key - arr[s]) / (arr[e] - arr[s])));

if (arr[position] == key){

// System.out.println("\n"+arr[position]);

return 1;

}

else if (arr[position]>key){

e = position-1;

}else {

s = position+1;

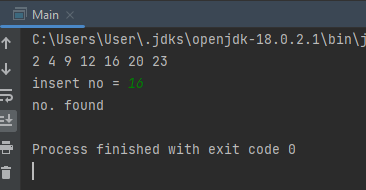
}

}

return -1;

}

}

}

**Question 02:**

import java.util.Scanner;

public class Question\_number2 {

public static void main(String[] args) {

int arr[] = new int[10];

Scanner sc = new Scanner(System.in);

int a = 47;

int m = 2743;

int b = 5943;

for (int i=0;i<10;i++){

arr[i] = ((a\*m)+(i\*b))%100000;

// arr[i] = temp%100000;

}

for (int i=0;i<10;i++){

System.out.println(arr[i]);

}

System.out.println("Enter number you wanna search: ");

int key = sc.nextInt();

int count =0;

int flag = 0;

int n = 0;

for (int i=0;i<10;i++){

count++;

if (key == arr[i]){

n = count;

flag = 1;

}

}

if (flag == 1){

System.out.println("Number appear after "+n+" comparisons..");

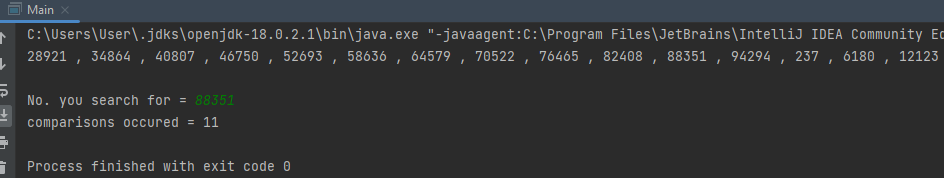
}else {

System.out.println("Number is not available in list.. ");

}

}

}



**Question 03:**

public class Main{

public static int findFirstOccurrence(int[] nums, int target){

// search space is nums[left…right]

int left = 0;

int right = nums.length - 1;

// initialize the result by -1

int result = -1;

// loop till the search space is exhausted

while (left <= right)

{

// find the mid-value in the search space and compares it with the target

int mid = (left + right) / 2;

// if the target is located, update the result and

// search towards the left (lower indices)

if (target == nums[mid])

{

result = mid;

right = mid - 1;

}

// if the target is less than the middle element, discard the right half

else if (target < nums[mid]) {

right = mid - 1;

}

// if the target is more than the middle element, discard the left half

else {

left = mid + 1;

}

}

// return the leftmost index, or -1 if the element is not found

return result;

}

public static int findLastOccurrence(int[] nums, int target)

{

// search space is nums[left…right]

int left = 0;

int right = nums.length - 1;

// initialize the result by -1

int result = -1;

// loop till the search space is exhausted

while (left <= right)

{

// find the mid-value in the search space and compares it with the target

int mid = (left + right) / 2;

// if the target is located, update the result and

// search towards the right (higher indices)

if (target == nums[mid])

{

result = mid;

left = mid + 1;

}

// if the target is less than the middle element, discard the right half

else if (target < nums[mid]) {

right = mid - 1;

}

// if the target is more than the middle element, discard the left half

else {

left = mid + 1;

}

}

// return the leftmost index, or -1 if the element is not found

return result;

}

public static void main(String[] args)

{

int[] nums = {2, 5, 5, 5, 6, 6, 8, 9, 9, 9};

int target = 5;

int index = findFirstOccurrence(nums, target);

if (index != -1)

{

System.out.println("The first occurrence of element " + target +

" is located at index " + index);

}

else {

System.out.println("Element not found in the array");

}

int index1 = findLastOccurrence(nums, target);

if (index1 != -1)

{

System.out.println("The last occurrence of element " + target +

" is located at index " + index1);

}

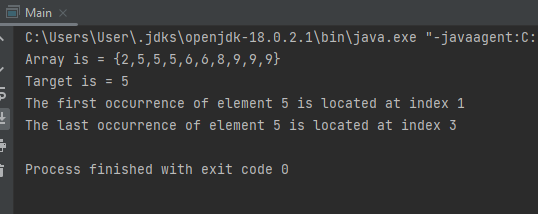
else {

System.out.println("Element not found in the array");

}

}

}



**Question 04:**

import java.util.\*;

public class Main{

public static int cards=26;

public static int s[]=new int[cards];

public static void q\_sort(int l,int h){

if(h-l<=0)

return;

else{

Random r=new Random();

int pi\_i=l+r.nextInt(h-l+1);

swapping(pi\_i,h);

int pivot=s[h];

int divide=division(l,h,pivot);

q\_sort(l,divide-1);

q\_sort(divide+1,h); }

}

public static int division(int l,int h,long pivot){

int lp=l-1;

int hp=h;

while(true){

while (s[++lp]<pivot) ;

while (hp>0&&s[--hp]>pivot) ;

if(lp>=hp)

break;

else

swapping(lp,hp); }

swapping(lp,h);

return lp;

}

public static void swapping(int i1,int i2){

int t=s[i1];

s[i1]=s[i2];

s[i2]=t;

}

static void display(int seq[]){

for(int j=0;j<seq.length;j++)

System.out.print(seq[j] + ",");

}

public static void main(String args[]){

Random random = new Random();

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

s[i]=Math.abs(random.nextInt(60));

System.out.print("\nRandom sequence before sort = ");

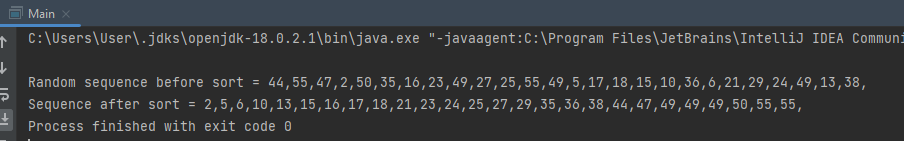
display(s);

System.out.print("\nSequence after sort = ");

q\_sort(0,cards-1);

display(s);

} }



**Question 05:**

import java.util.\*;

public class Q5 {

public static int Max(int array[], int n){

int temp = array[0];

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

if (array[i] > temp)

temp = array[i];

return temp;

}

public static void count(int array[], int n, int pos){

int output[] = new int[n];

int count[] = new int[10];

Arrays.fill(count, 0);

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

count[(array[i] / pos) % 10]++;

for (int i = 1; i < 10; i++)

count[i] += count[i-1];

for (int i = n - 1; i>= 0; i--) {

output[count[(array[i] / pos) % 10] - 1] = array[i];

count[(array[i] / pos) % 10]--;

}

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

array[i] = output[i];

}

public static void RadixSort(int arr[], int n){

int m = Max(arr, n);

for (int pos = 1; m / pos > 0; pos \*= 10)

count(arr, n, pos);

}

public static void print(int arr[], int n) {

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

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

}

public static void main(String[] args) {

int array[] = { 170, 45, 75, 90, 802, 24, 2, 66 };

int n = array.length;

RadixSort(array, n);

print(array, n);

}

}

