Code : 1  
package Topic\_04\_Arrays;

import java.util.\*;

public class A\_SpanOfArray {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

int n = s.nextInt();

int[] a = new int[n];

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

a[i] = s.nextInt();

}

int rv = findMaxandMin(a);

System.out.println(rv);

}

private static int findMaxandMin(int[] a) {

int rv = 0;

int max = a[0];

int min = a[0];

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

if (a[i] > max)

max = a[i];

if (a[i] < min)

min = a[i];

}

return max - min;

}

}

Code : 2  
package Topic\_04\_Arrays;

import java.util.Scanner;

public class B\_FindAnElementInArray {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

int n = s.nextInt();

int[] a = new int[n];

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

a[i] = s.nextInt();

}

int key = s.nextInt();

int rv = findElement(a, key);

System.out.println(rv);

}

private static int findElement(int[] a, int key) {

int rv = -1;

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

if (a[i] == key) {

rv = i;

break;

}

}

return rv;

}

}

Code : 3  
package Topic\_04\_Arrays;

import java.util.Scanner;

public class C\_BarChartUsingArrays {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

int n = s.nextInt();

int[] a = new int[n];

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

a[i] = s.nextInt();

}

printBarChart(a);

}

private static void printBarChart(int[] a) {

int max = findMax(a);

for (int i = max; i > 0; i--) {

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

if (i <= a[j]) {

System.out.print("\*\t");

} else {

System.out.print("\t");

}

}

System.out.println();

}

}

private static int findMax(int[] a) {

int max = a[0];

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

if (a[i] > max)

max = a[i];

}

return max;

}

}

Code : 4  
package Topic\_04\_Arrays;

import java.util.Scanner;

public class C\_InvertedBarChartUsingArrays {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

int n = s.nextInt();

int[] a = new int[n];

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

a[i] = s.nextInt();

}

printBarChart(a);

}

private static void printBarChart(int[] a) {

int max = findMax(a);

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

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

if (i < a[j]) {

System.out.print("\*\t");

} else {

System.out.print("\t");

}

}

System.out.println();

}

}

private static int findMax(int[] a) {

int max = a[0];

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

if (a[i] > max)

max = a[i];

}

return max;

}

}

Code : 5  
package Topic\_04\_Arrays;

import java.util.\*;

public class D\_SumOfTwoArrays {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

int n1 = s.nextInt();

int[] a = new int[n1];

for (int i = 0; i < n1; i++) {

a[i] = s.nextInt();

}

int n2 = s.nextInt();

int[] b = new int[n2];

for (int i = 0; i < n2; i++) {

b[i] = s.nextInt();

}

var c = sumOfArrays(a, b);

displayArray(c);

}

private static int[] sumOfArrays(int[] a, int[] b) {

int[] res = new int[(a.length > b.length ? a.length : b.length)];

int c = 0; // carry

for (int i = a.length - 1, j = b.length - 1, k = res.length - 1; k >= 0; i--, j--, k--) {

int d = c;

if (i >= 0)

d += a[i];

if (j >= 0)

d += b[j];

if (d > 9) {

res[k] = d % 10;

c = d / 10;

} else {

c = 0;

res[k] = d;

}

}

return res;

}

private static void displayArray(int[] c) {

for (int ele : c) {

System.out.println(ele);

}

}

}

Code : 6  
package Topic\_04\_Arrays;

import java.util.\*;

public class E\_DifferenceOfTwoArrays {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

int n1 = s.nextInt();

int[] a = new int[n1];

for (int i = 0; i < n1; i++) {

a[i] = s.nextInt();

}

int n2 = s.nextInt();

int[] b = new int[n2];

for (int i = 0; i < n2; i++) {

b[i] = s.nextInt();

}

var c = diffOfTwoArrays(a, b);

displayArray(c);

}

private static int[] diffOfTwoArrays(int[] a, int[] b) {

int[] res = new int[(a.length > b.length ? a.length : b.length)];

int borrow = 0;

for (int i = a.length - 1, j = b.length - 1, k = res.length - 1; k >= 0; i--, j--, k--) {

int d = 0;

d = (b[j]-borrow);

if(i>=0)

d=d-a[i];

if (d < 0) {

d = d + 10;// - borrow;

borrow = 1;

} else {

borrow = 0;

d = d;

}

res[k] = d;

}

return res;

}

private static void displayArray(int[] c) {

int i=0;

while(c[i]==0) {

i++;

}

while(i<c.length) {

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

i++;

}

}

}

Code : 7  
package Topic\_04\_Arrays;

import java.util.\*;

public class F\_ReverseOfArray {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

int n1 = s.nextInt();

int[] a = new int[n1];

for (int i = 0; i < n1; i++) {

a[i] = s.nextInt();

}

reverse(a);

displayArray(a);

}

private static void reverse(int[] a) {

int i = 0;

int j = a.length - 1;

while (i <= j) {

int temp = a[i];

a[i] = a[j];

a[j] = temp;

i++;

j--;

}

}

private static void displayArray(int[] c) {

int i = 0;

while (i < c.length) {

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

i++;

}

}

}

Code : 8  
package Topic\_04\_Arrays;

import java.util.\*;

public class G\_RotateAnArray {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

int n1 = s.nextInt();

int[] a = new int[n1];

for (int i = 0; i < n1; i++) {

a[i] = s.nextInt();

}

int k = s.nextInt();

rotate(a, k);

displayArray(a);

}

private static void rotate(int[] a, int k) {

k = k % a.length;

if (k < 0) {

k = k + a.length;

}

reverse(a, 0, a.length - 1);

reverse(a, 0, k - 1);

reverse(a, k, a.length - 1);

}

private static void reverse(int[] a, int left, int right) {

while (left <= right) {

int temp = a[left];

a[left] = a[right];

a[right] = temp;

left++;

right--;

}

}

private static void displayArray(int[] c) {

int i = 0;

while (i < c.length) {

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

i++;

}

}

}

Code : 9  
package Topic\_04\_Arrays;

import java.util.\*;

public class H\_InverseOfArray {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

int n1 = s.nextInt();

int[] a = new int[n1];

for (int i = 0; i < n1; i++) {

a[i] = s.nextInt();

}

a = inverse(a);

displayArray(a);

}

private static int[] inverse(int[] a) {

int i = 0;

int res[] = new int[a.length];

while (i < a.length) {

res[a[i]] = i;

i++;

}

return res;

}

private static void displayArray(int[] c) {

int i = 0;

while (i < c.length) {

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

i++;

}

}

}

Code : 10  
package Topic\_04\_Arrays;

import java.io.\*;

import java.util.\*;

public class I\_AllSubarrays {

public static void main(String[] args) throws Exception {

// write your code here

Scanner scn = new Scanner(System.in);

int n = scn.nextInt();

int[] arr = new int[n];

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

arr[i] = scn.nextInt();

}

PrintSubArrays(arr);

}

private static void PrintSubArrays(int[] arr) {

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

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

for (int k = i; k <= j; k++) {

System.out.print(arr[k] + "\t");

}

System.out.println();

}

}

}

}

Code : 11  
package Topic\_04\_Arrays;

import java.util.Scanner;

public class J\_SubSetsOfArray {

public static void main(String[] args) throws Exception {

// write your code here

Scanner scn = new Scanner(System.in);

int n = scn.nextInt();

int[] arr = new int[n];

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

arr[i] = scn.nextInt();

}

SubSetOfArrays(arr);

}

private static void SubSetOfArrays(int[] a) {

int p = (int) Math.pow(2, a.length);

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

int temp = i;

String set = "";

for (int j = a.length - 1; j >= 0; j--) {

int rem = temp % 2;

temp = temp / 2;

if (rem == 0) {

set = "-\t" + set;

} else {

set = a[j] + "\t" + set;

}

}

System.out.println(set);

}

}

}

Code : 12  
package Topic\_04\_Arrays;

import java.io.\*;

import java.util.\*;

public class K\_BrokenEconomy\_FloorAndCeil{

public static void main(String[] args) throws Exception {

// write your code here

Scanner scn = new Scanner(System.in);

int n = scn.nextInt();

int[] arr = new int[n];

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

arr[i] = scn.nextInt();

}

int data = scn.nextInt();

int left = 0;

int right = arr.length - 1;

int floor = -1; // the greatest among smaller numbers

int ceil = -1; // the smallest among greater numbers

while (left <= right) {

int mid = (left + right) / 2;

if (data > arr[mid]) {

left = mid + 1;

// left to mid are all small and mid is greatest of them

floor = arr[mid];

} else if (data < arr[mid]) {

right = mid - 1;

// mid to right are all greater and mid is the smallest of them

ceil = arr[mid];

} else {

floor = arr[mid];

ceil = arr[mid];

break;

}

}

System.out.println(ceil);

System.out.println(floor);

}

}

Code : 13  
package Topic\_04\_Arrays;

import java.util.\*;

public class L\_FiAndLi{

public static void main(String[] args) {

Scanner scn = new Scanner(System.in);

int n = scn.nextInt();

int[] arr = new int[n];

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

arr[i] = scn.nextInt();

}

int data = scn.nextInt();

int left = 0;

int right = arr.length - 1;

int fi = -1;

while (left <= right) {

int mid = (left + right) / 2;

if (data > arr[mid]) {

// left side is useless, discard it

left = mid + 1;

} else if (data < arr[mid]) {

// right side is useless, discard it

right = mid - 1;

} else {

fi = mid;

right = mid - 1;

}

}

System.out.println(fi);

int li = -1;

left = 0;

right = arr.length - 1;

while (left <= right) {

int mid = (left + right) / 2;

if (data > arr[mid]) {

// left side is useless, discard it

left = mid + 1;

} else if (data < arr[mid]) {

// right side is useless, discard it

right = mid - 1;

} else {

li = mid;

left = mid + 1;

}

}

System.out.println(li);

}

}

Code : 14  
package Topic\_04\_Arrays;

import java.util.\*;

public class Z\_BinarySearch {

public static void main(String[] args) {

Scanner scn = new Scanner(System.in);

int n = scn.nextInt();

int[] arr = new int[n];

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

arr[i] = scn.nextInt();

}

int data = scn.nextInt();

int left = 0;

int right = arr.length - 1;

int foundAt = -1;

while (left <= right) {

int mid = (left + right) / 2;

if (data > arr[mid]) {

// left side is useless, discard it

left = mid + 1;

} else if (data < arr[mid]) {

// right side is useless, discard it

right = mid - 1;

} else {

foundAt = mid;

break;

}

}

System.out.println(foundAt);

}

}