**WEEK 2**

WEEK 2.1: COUNT THE OCCURENCES

Problem Statement:

You are given an array of integers nums.You are also given an integer original

which is the first number that needs to be searched for in nums.

You then do the following steps:

If original is found in nums, multiply it by two (i.e., set original = 2 \* original).

Otherwise, stop the process.

Repeat this process with the new number as long as you keep finding the number.

Return the final value of original.

Input Format

First Line :The Array size n

Second Line : Array elements one in each line

Third Line : Original number

Output Format

First Line : the number

Sample Input

5

5 3 6 1 12

3

Sample Output

24

Explanation:

- 3 is found in nums. 3 is multiplied by 2 to obtain 6.

- 6 is found in nums. 6 is multiplied by 2 to obtain 12.

- 12 is found in nums. 12 is multiplied by 2 to obtain 24.

- 24 is not found in nums.Thus, 24 is returned.

PROGRAM:

import java.util.\*;

public class Java\_2\_1{

public static int rec(int arr[], int x){

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

if(arr[i]==x){

x = x\*2;

rec(arr, x);

}

}

return x;

}

public static void main(String args[]){

Scanner obj = new Scanner(System.in);

System.out.print("Enter the test cases: ");

int n = obj.nextInt();

int arr[] = new int[n];

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

System.out.printf("Enter the %dth element: ", i);

arr[i] = obj.nextInt();

}

System.out.print("Enter the target element: ");

int x = obj.nextInt();

int res = rec(arr, x);

System.out.print(res);

}

}

WEEK 2.2: INVENTORY MANAGEMENT

Problem Statement:

You are given an array prices where prices[i] is the price of a given stock on the

ith day.You want to maximize your profit by choosing a single day to buy one

stock and choosing a different day in the future to sell that stock.

Return the maximum profit you can achieve from this transaction. If you cannot

achieve any profit, return 0.

Input Format

First line :The array size : 6

Second Line :the array element s : 7 1 5 3 6 4

Output Format

First Line : 5

Sample Input

6

7 1 5 3 6 4

Sample Output

5

Explanation:

Buy on day 2 (price = 1) and sell on day 5 (price = 6), profit = 6-1 = 5.

Note that buying on day 2 and selling on day 1 is not allowed because you must

buy before you sell.

PROGRAM:

import java.util.\*;

public class Buy\_Sell{

public static void main(String args[]){

Scanner obj = new Scanner(System.in);

System.out.print("Enter the no. of elements: ");

int n = obj.nextInt();

int arr[] = new int[n];

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

System.out.printf("Enter the %dth element: ",i+1);

arr[i] = obj.nextInt();

}

int buy=arr[0];

int pivot=0;

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

if(arr[i]<buy){

buy = arr[i];

pivot = i;

}

}

int sell=pivot;

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

if(arr[i]>sell){

sell = arr[i];

}

}

System.out.printf("%d",sell-buy);

}

}

WEEK 2.3: SORT AN ARRAYS OF 0’s, 1’s AND 2’s

Problem Statement:

Given an Array of N with the elements of 0's, 1's and 2's.

Your task is to arrange the array elements in the following order.

0's followed by 1's followed 2's

Input Format

First Line : The Array size : n

Second Line: The array elements

Output Format

Print the array elements as expected.

Sample Input 1

6

{0, 1, 2, 0, 1, 2}

Sample Output 1

{0, 0, 1, 1, 2, 2}

Explanation: {0, 0, 1, 1, 2, 2} has all 0s first, then all 1s and all 2s in last.

Input: {0, 1, 1, 0, 1, 2, 1, 2, 0, 0, 0, 1}

Output: {0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 2, 2}

Explanation: {0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 2, 2} has all 0s first, then all 1s and all

2s in last.

PROGRAM:

import java.util.\*;

public class Sort{

public static void main(String args[]){

Scanner obj = new Scanner(System.in);

System.out.print("Enter the no. of elements: ");

int n = obj.nextInt();

int arr[] = new int[n];

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

System.out.printf("Enter the %dth element: ",i+1);

arr[i] = obj.nextInt();

}

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

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

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

int temp = arr[j+1];

arr[j+1] = arr[j];

arr[j] = temp;

}

}

}

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

System.out.printf("%d",arr[i]);

}

}

}

WEEK 2.4: FIND THE MISSING NUMBER

Problem Statement:

Given an array arr[] of size N-1 with integers in the range of [1, N], the task is to find

the missing number from the first N integers.

Note: There are no duplicates in the list.

Input Format

First Line : The array size : N

Second Line : The array elements

Output Format

Print the missing number .

SAMPLE INPUT 1

8

1 2 4 6 3 7 8

SAMPLE OUTPUT 1

5

Input: arr[] = {1, 2, 4, 6, 3, 7, 8} , N = 8

Output: 5

Explanation: Here the size of the array is 8, so the range will be [1, 8]

. The missing number between 1 to 8 is 5

Input: arr[] = {1, 2, 3, 5}, N = 5

Output: 4

Explanation: Here the size of the array is 4, so the range will be [1, 5].

The missing number between 1 to 5 is 4

PROGRAM:

import java.util.\*;

public class Missing{

public static void main(String args[]){

Scanner obj = new Scanner(System.in);

System.out.print("Enter the value: ");

int n = obj.nextInt();

int arr[] = new int[n];

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

System.out.printf("Enter Value %d: ",i+1);

arr[i] = obj.nextInt();

}

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

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

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

int temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

}

}

}

int flag = 0;

for(int i=0; i<n-2; i++){

if((arr[i+1]-arr[i])!=1){

flag = 1;

System.out.printf("%d",arr[i]+1);

}

else

continue;

}

if(flag==0)

System.out.print("1");

}

}

WEEK 2.5: MOVE ALL THE ZEROES TO THE END OF AN ARRAY

Problem Statement:

Given an array of N elements,You task is to move the Zeroes to the end of the Array.

Input Format

First Line : Array Size : N

Second Line: Array elements separated by space

Output Format

First line: Array elements arranged as zeroes at the end.

SAMPLE INPUT

5

1 0 2 0 3 0 4 0

SAMPLE OUTPUT

1 2 3 4 0 0 0

PROGRAM:

import java.util.\*;

public class Move\_Zeroes{

public static void main(String args[]){

Scanner obj = new Scanner(System.in);

System.out.print("Enter the no. of elements: ");

int n = obj.nextInt();

int arr[] = new int[n];

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

System.out.printf("Enter Value %d: ",i+1);

arr[i] = obj.nextInt();

}

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

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

if(arr[j]==0){

int temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

}

}

}

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

System.out.printf("%d ",arr[i]);

}

}

}