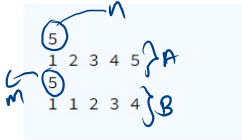


Double Occurrence

Given an array of size n with unique integer elements. And then take m as an integer input. Declare the **second array** of size m that stores values of int data-type. Then take m integer inputs and store them in the array one by one.

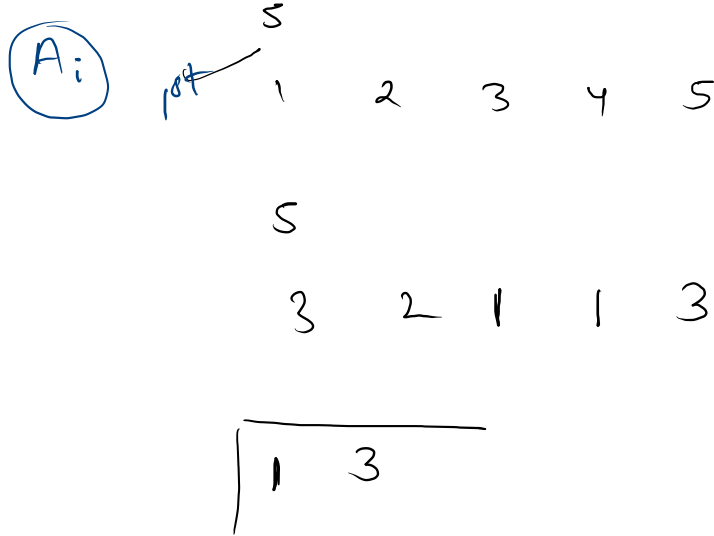
Then print all the elements of the first array which occur (exactly twice) in the second array.

Sample Input 0



Sample Output 0

1



5

1st → 1 2 3 4 5

as
—
2

7

2nd → 2 4 3 1 1 1 2

$n = 5$
A \rightarrow 1 2 (3) 4 5

$m = 8$
B \rightarrow 1 1 2 3 3 2 2 4

$\textcircled{3} - 1 : \textcircled{2} = 2$
 $\hookrightarrow \text{print } \textcircled{1}$

$2 : 3 \neq 2$

$3 : \textcircled{2} = 2$
 $\hookrightarrow \text{print } \textcircled{2}$

func

(Array, int)
x

x = 3
↓

3	3	1	2	3	4
---	---	---	---	---	---

adv.

```

3
4 public class Solution {
5     public static int countOcc(int [] arr, int x){
6         int count = 0;
7         for(int i = 0; i < arr.length; i++){
8             if(arr[i] == x){
9                 count++;
10            }
11        }
12        return count;
13    }
14
15    public static void main(String[] args) {
16        Scanner scn = new Scanner(System.in);
17        int n = scn.nextInt();
18        int [] A = new int[n];
19        for(int i = 0; i < n; i++){
20            A[i] = scn.nextInt();
21        }
22        int m = scn.nextInt();
23        int [] B = new int[m];
24        for(int i = 0; i < m; i++){
25            B[i] = scn.nextInt();
26        }
27
28        //logic
29        for(int i = 0; i < n; i++){
30            int x = A[i];
31            int count = countOcc(B, x);
32            if(count == 2){
33                System.out.print(x + " ");
34            }
35        }
36
37    }
38 }

```

A →

1	2	3	4	5
<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>

B →

4	3	5	4	1	2	3
<u>4</u>	<u>3</u>	<u>5</u>	<u>4</u>	<u>1</u>	<u>2</u>	<u>3</u>

$x = A[i] = A[3] = 4$

$x = 4$

$2 == 2$

↪

$\sqrt{3 \ 4}$

A →

1	2	3	4	5
---	---	---	---	---

* count how many times (x)
occuring in B
array.

B →

4	3	5	4	2	1	3
---	---	---	---	---	---	---

↓
count == 2
↳ print

maximum difference between the two elements

Given an array `arr[]` of integers, find out the maximum difference between any two elements such that larger element appears after the smaller number

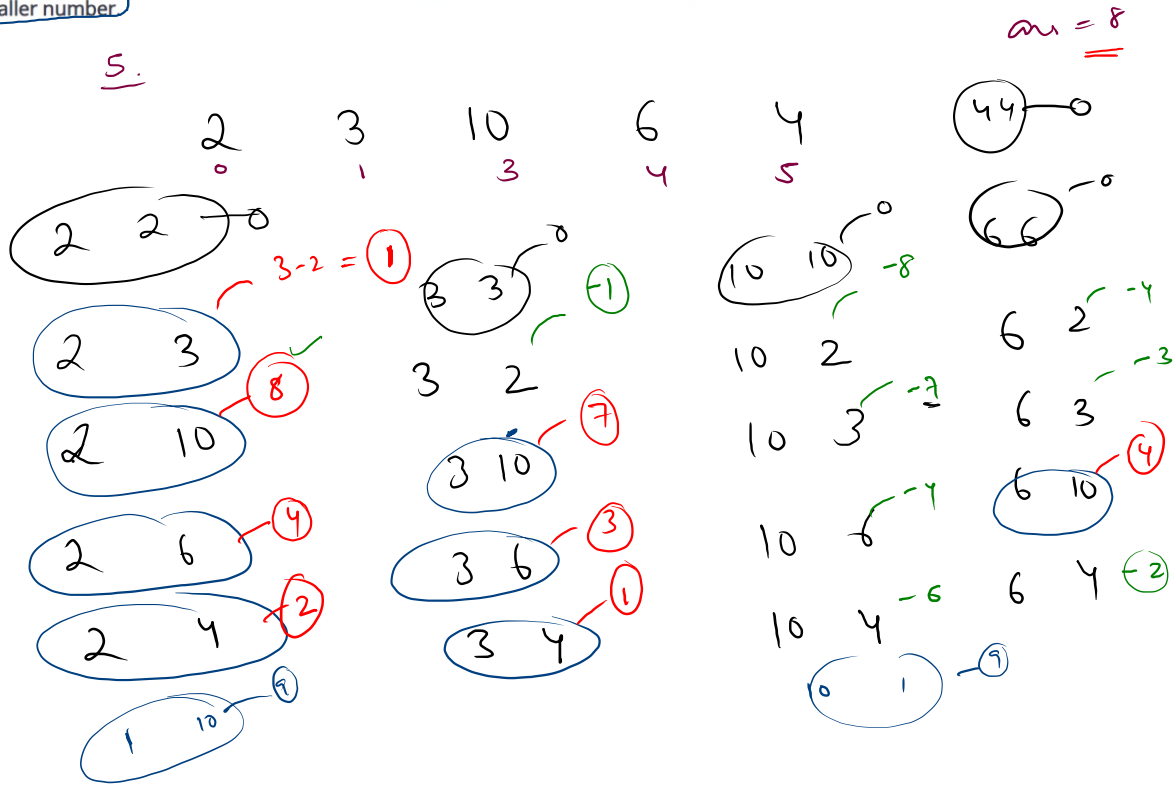
Sample Input 0

```
7
2 3 10 6 4 8 1
```

Sample Output 0

```
8
```

find max diff of all pair.



Sample Input 0

```

7
2 3 10 5 4 8 1
8 1 2 3 4 5 6

```

Sample Output 0

```

8

```

8-L

$$8-L = 2-8$$

$$8-2$$

S (L)

2 8

$$\begin{array}{l}
 2 \quad 3_1 = 1 \quad 1 \quad 3 \quad 10_2 = 7 \quad 2 \quad 10 \quad 6 = -7 \quad 5 \quad 6 \quad 7_4 = -2 \\
 2 \quad 10_2 = 8 \quad 1 \quad 3 \quad 6_3 = 4 \quad 2 \quad 10 \quad 4_5 = -6 \quad 3 \quad 6 \quad 8 = +2 \\
 2 \quad 6_3 = 4 \quad 1 \quad 3 \quad 4_4 = 1 \quad 2 \quad 10 \quad 8_5 = -2 \quad 3 \quad 6 \quad 1_5 = \textcircled{5} \\
 2 \quad 4_4 = 2 \quad 1 \quad 3 \quad 8_5 = 5 \quad 2 \quad 10 \quad 1_6 = \textcircled{-9} \\
 2 \quad 8_5 = 6 \quad 1 \quad 3 \quad 1_6 = -2 \\
 2 \quad 1_6 = -1
 \end{array}$$

$$\begin{array}{l}
 4 \quad 4 \quad 8 = 7 \quad 5 \quad 8 \quad 1_6 = -7 \\
 4 \quad 4 \quad 1 = -3 \quad 6
 \end{array}$$

0 2 \rightarrow i+1

0	1	2	3	4	5	6
1	2	3	4	5	6	
2	3	4	5	6		
3	4	5	6			
4	5	6				
5	6					

$$d = -\infty$$

$$n = 4$$

```

1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner scn = new Scanner(System.in);
8         int n = scn.nextInt();
9         int [] A = new int[n];
10        for(int i = 0; i < n; i++){
11            A[i] = scn.nextInt();
12        }
13
14        int d = Integer.MIN_VALUE;
15        //logic
16        for(int i = 0; i < n; i++){
17            for(int j = i+1; j < n; j++){
18                d = Math.max(d, A[j] - A[i]);
19            }
20        }
21        System.out.println(d);
22    }
23 }

```

2	6	1	3	
0	1	2	3	∞
				j =

$$i = 0$$

$$0 < 4 \checkmark$$

$$1 - 2 = (-1)$$

$$j = 1$$

$$1 < 4 \checkmark$$

$$2$$

$$2 < 4 \checkmark$$

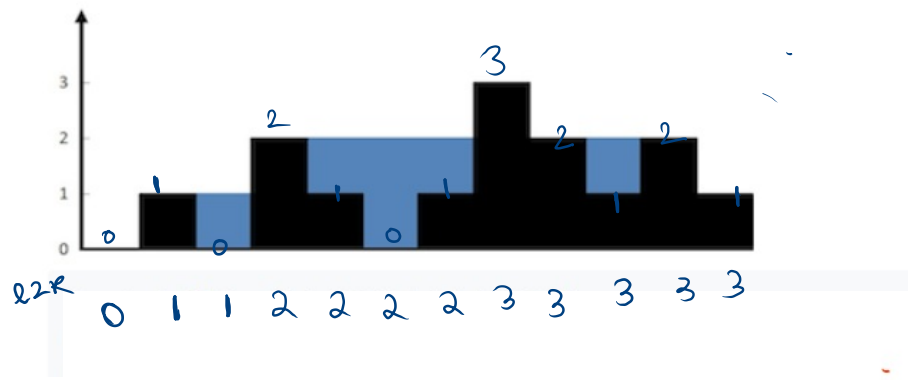
$$3 - 2 = 1$$

Store Maximum

Given n non-negative integers representing an elevation map where the width of each bar is 1, compute how much **maximum water** it can trap after raining.

$$\min(L, R)$$

$$(3, 0) = 0 - 0 = 0$$



L
↓
max till me from left

R
↓
max till me from right

l2r \rightarrow max till me from left
 r2l \rightarrow max till me from right

min (l2r, r2l) - A[i]

A

0	1	0	2	1	0	1	3	2	1	2	1
0	1	2	3	4	5	6	7	8	9	10	11

l2r

0	1	1	2	2	2	2	3	3	3	3	3
3	3	3	3	3	3	3	3	2	2	2	1

r2l

ans \rightarrow 0 0 1 0 1 2 1 0 0 1 0 0 ans=6

$$ans += \min(l2r[i], r2l[i]) - A[i]$$

