

Find the number!

Given an unsorted array of n elements. find if the element is present in the array or not
Complete the find Number function in the editor below. It has 2 parameters

1. An array of integers, arr denoting the elements in the array
2. An integer, K , denoting the element to be searched in the array

The function must return a string "YES" or "NO" denoting if the element is present in the array or not.

Input Format :

The first line contains an integer n , denoting the number of elements in the array arr . Each line of the n subsequent lines contains an integer describing arr . The next line contains an integer k the element that needs to be searched.

Constraints :

- $1 \leq n \leq 10^5$
- $1 \leq arr[i] \leq 10^9$

Output Format :

The function must return a string "YES" or "NO" denoting if the element is present in the array or not. This is printed to stdout by locked stub

code in the editor

Sample Input :

```
5
1
2
3
4
5
1
```

Sample Output :

YES

Explanation :

Given the array $[1, 2, 3, 4, 5]$ we want to find the element 1 if it is present or not. We can find the element 1 at index 0. Therefore we print "YES"

Code :

```
import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.function.*;
import java.util.regex.*;
import java.util.stream.*;
```

```
import static java.util.stream.Collectors.joining;
```

```
import static java.util.stream.Collectors.toList;
```

```
class Result {
```

```
/*
```

```
Complete the findNumber function below.
```

The function is expected to return a STRING.

The function accepts following parameters:

1. INTEGER ARRAY arr * 2. INTEGER k

```
*/
```

```
public static String findNumber (List<Integer> arr, int k) {
```

```
// Write your code here
```

```
}
```

```
}
```

```
public class Solution{
```

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

```
BufferedReader bufferedReader = new BufferedReader (new InputStreamReader (System.in));
```

```
BufferedWriter bufferedWriter = new BufferedWriter(new
FileWriter(System.getenv("OUTPUT_PATH")));
```

```
int arrCount = Integer.parseInt(bufferedReader.readLine().trim());
```

```

List<Integer> arr = IntStream.range(e, arrCount).mapToObj (i -> {
try {

return bufferedReader.readLine().replaceAll("\\s+$", "");

} catch (IOException ex) {

throw new RuntimeException(ex);
}
}}

.map(String:: trim)

.map(Integer::parseInt)

.collect(toList());

int k = Integer.parseInt(bufferedReader.readLine().trim());

String result = Result. findNumber (arr, k);

bufferedWriter.write(result); bufferedWriter.newLine();

bufferedReader.close();
bufferedWriter.close();
}
}

```