Given a string you need to print the size of the longest possible substring that has exactly K unique characters. If there is no possible substring then print -1.

Input Format

Take the String S as in input

Take the Integer K as in input

Constraints

1 ≤ |S| ≤ 10^5

1 ≤ K ≤ 26

Output Format

returns the length of the longest substring with exactly K distinct characters. If there is no substring with exactly K distinct characters then return -1.

Sample Input 0

```
aabacbebebe
3
```

Sample Output 0

```
7
```

Sample Input 1

```
aaaa
2
```

Sample Output 1

```
-1
```

```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String str = sc.nextLine();
        int k = sc.nextInt();
        int max = -1;

        for (int i = 0; i < str.length(); i++) {
            int j = i;
            int count = 0;
            HashSet<Character> st = new HashSet<>();

            while (j < str.length()) {
                char ch = str.charAt(j);
                st.add(ch);

                if (st.size() > k) {
                    break;
                }

                if (st.size() == k) {
                    count = j - i + 1;
                    max = Math.max(max, count);
                }

                j++;
            }
        }

        System.out.println(max);
    }
}
```

Check if array pair are divisible by K

HW_Pair that is divisible by 60

You are given a list of songs where the ith song has a duration of time[i] seconds.

Return the number of pairs of songs for which their total duration in seconds is divisible by 60. Formally, we want the number of indices i, j such that i < j with (time[i] + time[j]) % 60 == 0.

Input Format

Take the Integer N as size of an array

Take the N integer as element of an array

Constraints

1 <= time.length <= 6 * 10^4

1 <= time[i] <= 500

Output Format

Return the number of pairs of songs for which their total duration in seconds is divisible by 60.

Sample Input 0

```
5
30
20
150
100
40
```

Sample Output 0

```
3
```

Explanation 0

```
Three pairs have a total duration divisible by 60:
(time[0] = 30, time[2] = 150): total duration 180
(time[1] = 20, time[3] = 100): total duration 120
(time[1] = 20, time[4] = 40): total duration 60
```

60 / 60 = 1

30 , 20 , 150 , 100 , 40

30 + 30 = 60

60 - 30 = 30

20 + 40 = 60

150 + 30 = 180

20 + 100 = 120

20 + 40 = 60

30 % 60 = 30

60 - 30 = 30

20 % 60 = 20

150 % 60 = 30

20 % 60 = 20

need = 60 - 20

need = 40

40 | 1

20

Submitted Code

Language: Java 8

```
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int arr[] = new int[n];
        for(int i=0;i<n;i++)arr[i]=sc.nextInt();

        Map<Integer,Integer> hm = new HashMap<>();
        int count =0;
        for(int i =0;i<n;i++){
            int remain = arr[i] %60;
            int need =remain==0?0:60-remain;
            if(hm.containsKey(need))count+=hm.get(need);

            hm.put(remain,hm.getOrDefault(remain,0)+1);
        }
        System.out.print(count);
    }
}
```

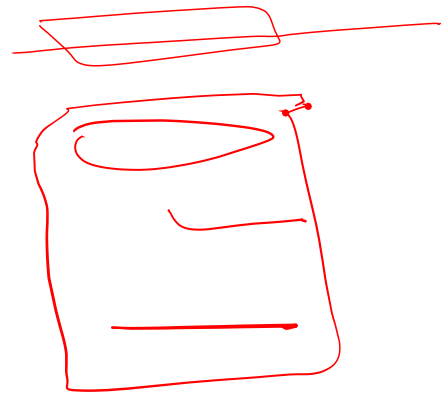
count = 3

key	value
30	2
20	1
40	2

remain =

need =

HW_Queue Reversal 5



Given a Queue Q containing N elements. The task is to reverse the Queue.

Input Format

The First line integer N represent the size of Queue.

The Second line N integer represent the element of Queue.

Constraints

1 ≤ N ≤ 10^5

1 ≤ elements of Queue ≤ 10^5

Output Format

returns the reversed queue.

Sample Input 0

```
5
1 2 3 5 6
```

Sample Output 0

```
6 5 3 2 1
```

Submitted Code

Language: Java 8

```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner sc= new Scanner(System.in);
        int n = sc.nextInt();
        Queue<Integer> qe = new LinkedList<>();
        for(int i=0;i<n;i++)qe.add(sc.nextInt());

        Stack<Integer> st = new Stack<>();
        while(!qe.isEmpty()){
            st.add(qe.remove());
        }
        while(!st.isEmpty()){
            System.out.print(st.pop()+" ");
        }
    }
}
```