

阿里巴巴找黄金宝箱(III)题目描述：

一贫如洗的樵夫阿里巴巴在去砍柴的路上，无意中发现了强盗集团的藏宝地，藏宝地有编号从  $0 \sim N$  的箱子，每个箱子上面贴有一个数字。

阿里巴巴念出一个咒语数字，查看宝箱是否存在两个不同箱子，这两个箱子上贴的数字相同，同时这两个箱子的编号之差的绝对值小于等于咒语数字，如果存在这样的一对宝箱，请返回最先找到的那对宝箱左边箱子的编号，如果不存在则返回  $-1$ 。

输入描述：

第一行输入一个数字字符串，数字之间使用逗号分隔，例如：1,2,3,1

字符串中数字个数  $\geq 1$ ， $\leq 100000$ ；每个数字值  $\geq -100000$ ， $\leq 100000$ ；

第二行输入咒语数字，例如：3，咒语数字  $\geq 1$ ， $\leq 100000$

输出描述：

存在这样的一对宝箱，请返回最先找到的那对宝箱左边箱子的编号，如果不存在则返回  $-$

示例 1

输入：

6,3,1,6

3

输出：

0

说明：

示例 2

输入：

5,6,7,5,6,7

2

输出：

-1

说明：

```
import java.util.ArrayList;
```

```
import java.util.HashMap;
```

```
import java.util.List;
```

```
import java.util.Scanner;
```

```
// 注意类名必须为 Main, 不要有任何 package xxx 信息
```

```
public class Main {
```

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

```
        Scanner in = new Scanner(System.in);
```

```
        // 注意 hasNext 和 hasNextLine 的区别
```

```
        while (in.hasNextLine()) { // 注意 while 处理多个 case
```

```
            String a = in.nextLine();
```

```
            int b = Integer.parseInt(in.nextLine());
```

```
            String[] s = a.split(",");
```

```
            int[] temp = new int[s.length];
```

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

```
                temp[i]=Integer.parseInt(s[i]);
```

```
            }
```

```
            HashMap<Integer, List<Integer>> map = new HashMap<>();
```

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

```
int j = temp[i];
```

```
List<Integer> list= map.getDefault(j, new
```

```
ArrayList<Integer>());
```

```
list.add(i);
```

```
map.put(j,list);
```

```
}
```

```
int min = -1;
```

```
for (Integer integer : map.keySet()) {
```

```
List<Integer> list = map.get(integer);
```

```
if (list!=null&&list.size()>1){
```

```
for (int i = 0; i < list.size()-1; i++) {
```

```
Integer pre = list.get(i);
```

```
Integer next = list.get(i+1);
```

```
int k = next - pre;
```

```
if (k<=b){
```

```
if (min== -1){
```

```
min=list.get(i);
```

```
}else {
```

```
min=Math.min(min,list.get(i));
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
System.out.println(min);
```

```
}
```

```
}
```

```
}
```