

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~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  $k(k < N)$ ，找出连续  $k$  个宝箱数字和的最大值，并输出该最大值。

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: 2,10,-3,-8,40,5

字符串中数字的个数  $\geq 1$ ， $\leq 100000$ ；每个数字  $\geq -10000$ ， $\leq 10000$ ；

第二行输入咒语数字，例如：4，咒语数字大小小于宝箱的个数

输出描述：

连续  $k$  个宝箱数字和的最大值，例如：39

补充说明：

示例1

输入：2,10,-3,-8,40,5

4

输出：39

说明：

示例2

输入：8

1

输出：8

说明：

```

1  import java.util.*;
2
3  // 注意类名必须为 Main, 不要有任何 package xxx 信息
4  public class Main {
5      public static void main(String[] args) {
6          Scanner in = new Scanner(System.in);
7          // 注意 hasNext 和 hasNextLine 的区别
8          while (in.hasNext()) { // 注意 while 处理多个 case
9              String t = in.nextLine();
10             int[] nums = new int[100001];
11             nums[0]=0;
12             String[] numst = t.split(",");
13             for(int i=0;i<numst.length;i++){
14                 nums[i+1]=Integer.valueOf(numst[i]);
15                 nums[i+1]= nums[i+1]+nums[i];
16             }
17             int k = in.nextInt();
18             int slow=0,fast=k;
19             int result = Integer.MIN_VALUE;
20             int len = numst.length+1;
21             while (fast<len){
22                 if(nums[fast]-nums[slow]>result){
23                     result=nums[fast]-nums[slow];
24                 }
25                 slow++;
26                 fast++;
27             }
28             System.out.println(result);
29         }
30     }
31 }

```