

JAVA-数组-贪吃的猴子-一只贪吃的猴子来到一个果园

题目描述：

一只贪吃的猴子，来到一个果园，发现许多串香蕉排成一行，每串香蕉上有若干根香蕉。每串香蕉的根数由数组 `numbers` 给出。猴子获取香蕉，每次都只能从行的开头或者末尾获取，并且只能获取 `N` 次，求猴子最多能获取多少根香蕉。

输入描述：

第一行为数组 `numbers` 的长度

第二行为数组 `numbers` 的值每个数字通过空格分开

第三行输入为 `N`，表示获取的次数

输出描述：

按照题目要求能获取的最大数值

补充说明：

$1 \leq \text{numbers.length} \leq 100000$

$1 \leq \text{numbers}[i] \leq 100$

$1 \leq N \leq \text{numbers.length}$

示例 1

输入：

7

1 2 2 7 3 6 1

3

输出：

10

说明：

第一次获取香蕉，无论是从行的开头或者末尾获取，得到的香蕉根数目为 1。

但是，从行末尾获取能获取到最优的策略，后面可以直接得到香蕉根数目 6 和 3。

因此最终根数为 $1+6+3 = 10$

示例 2

输入：

3

1 2 3

3

输出：

6

说明：

全部获取所有的香蕉，因此最终根数为 $1+2+3 = 6$

示例 3

输入：

4

4 2 2 3

2

输出：

7

说明：

第一次获取香蕉为行的开头，第二次获取为行的末尾，因此最终根数为 $4+3 = 7$

```
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.hasNextInt()) { // 注意 while 处理多个 case  
            int n = in.nextInt();  
            int a[]=new int[n];  
            int left=0;  
            int right=0;  
            int max=0;  
            int sum=0;  
            for(int i=0;i<n;i++)  
            {  
                a[i]=in.nextInt();  
            }  
            int m=in.nextInt();  
            for(int i=0;i<=m;i++)  
            {  
                sum=0;  
                left=i;  
                right=m-left;  
                for(int c=0;c<left;c++)  
                {  
                    sum=sum+a[c];  
                }  
                for(int d=a.length-right;d<a.length;d++)  
                {  
                    sum=sum+a[d];  
                }  
                max=Math.max(max,sum);  
            }  
            System.out.print(max);  
        }  
    }  
}
```