

面试 interview

题目描述:

A 为面试官, B 表示 s_z_1 同学。

A: “s_z_1 同学, 我们看了你的成绩, 都还不错, 但为什么 apio 的成绩是铜牌呢?”

B: “(◎_◎)…因为前一天没休息好。”(显然是因为考场上逗逼)

A: “嗯, 不过你这次的测试成绩很让我们满意。这样吧, 我们来问你一道题。”

B: “O(∩_∩)O 好的。”(我能说不吗)

A: “题目是这样的: 袋子里有三种颜色的小球, 分别为红绿蓝 ……”

B: (完了, 这是要考 polya 定理的节奏啊! Burnside 引理什么的我最弱了啊, 求轻虐
~~~~(>\_<)~~~~)

A: “红色 4 个, 绿色 6 个, 蓝色 7 个 ……”

B: (速度记下来, 马上该求各种置换, 各种组合了)

A: “如果想得到 5 个相同颜色的小球, 最少拿几个”

B: (抬起头呆呆的望着面试官。你 tmd 在逗我笑吧)

下一个面试的就该是你了, 为了熟练掌握这种类型的题目。你决定随机造一些数据来考自己。即一个袋子里装着  $n$  种不同颜色的小球, 每种颜色的小球有不同的数量, 你想知道, 每次可以拿出一个小球并且不放回去的前提下, 想得到  $K$  个相同颜色的小球, 最少拿多少次。

## 输入格式:

输入共两行, 第一行两个整数  $n, K$ , 表示含义如题。

第二行  $n$  个整数  $a[i]$ , 表示第  $i$  种颜色有  $a[i]$  个小球。

## 输出格式:

一行一个数表示最少拿多少次, 如果做不到, 输出 “Impossible!” (不含引号)。

## 样例输入(interview.in):

3 5  
4 6 7

## 样例输出(interview.out):

13

## 数据范围:

100%的数据:  $1 \leq n \leq 10^6$ ,  $0 \leq K \leq 10^9$ ,  $1 \leq a[i] \leq 10^9$

# 软妹币 rmb

## 题目描述:

一日,大土豪 prey 碰见了 s\_z\_1,望着 s\_z\_1 手上几张皱巴巴的第一代软妹币,为了世界的和谐共生,说道:“看看你手中可怜的几张软妹币,在这个物价飞涨的年代,根本不够花好嘛。真不知道你是怎么像水熊虫一样活下来的。这样吧,为了展现我等善良友好的风貌,我决定:咳咳,用你现在手中所有的软妹币,如果能凑出 1 到 m 中的所有面值,我就给你  $m(m+1)/2$  元的奖励。好了,我还有事,你去找我的秘书吧!”

于是 s\_z\_1 找到了你-----大土豪的秘书!

## 输入格式:

输入共三行,第一行一个整数 n,表示当前国家发行的软妹币面值种类数。  
第二行 n 个整数, a[i] 表示第 i 种钱币的面额,保证  $a[1] < a[2] < a[3] < \dots < a[n]$ 。  
第三行 n 个整数, b[i] 表示第 i 中面额的钱币 s\_z\_1 拥有的个数。

## 输出格式:

一行一个数表示你应付给 s\_z\_1 多少钱(反正不是你的钱,不给白不给!)

## 样例输入(rmb.in):

```
4
1 2 5 10
2 2 1 0
```

## 样例输出(rmb.out):

```
66
```

## 数据范围:

30%的数据:  $1 \leq n \leq 100$ ,  $1 \leq a[i] \leq 100$ ,  $0 \leq b[i] \leq 100$   
60%的数据:  $1 \leq n \leq 100$ ,  $1 \leq a[i] \leq 10^5$ ,  $0 \leq b[i] \leq 100$   
100%的数据:  $1 \leq n \leq 100$ ,  $1 \leq a[i] \leq 10^{100}$ ,  $0 \leq b[i] \leq 100$

## 样例解释:

```
1 = 1
2 = 2
3 = 1 + 2
4 = 2 + 2
5 = 5
6 = 1 + 1 + 2 + 2
7 = 5 + 1 + 1
8 = 5 + 2 + 1
9 = 5 + 2 + 2
10 = 5 + 2 + 2 + 1
11 = 5 + 2 + 2 + 1 + 1
Ans = (11+1) * 11 / 2 = 66
```

# 踢毽子 shuttlecock

## 题目描述:

自从上高三以来，s\_z\_1和他的小伙伴们喜欢上了踢毽子（不要问为什么，等你上高三就明白了O\_\_O“……”）。每天下午，趁着学弟学妹们跑操的时候，全班的n位同学，就会来到操场上，分成了很多组踢毽子。每组同学都站成一个圆圈，因为技术性问题，每位同学能且仅能在接到毽子后将它传给左右两位小伙伴之一。可有的时候，有些同学会蹲下系鞋带，那他就不能参加游戏喽。当然，等他系好鞋带后，又会重新站起来加入到游戏中去。

此时的你因为受伤，坐在主席台旁，看着下面愉快地玩耍的小伙伴们，心中感慨万千。很快你发现，在每个时刻，都可能有人改变状态，即从站着到蹲下系鞋带，或者系好鞋带重新站起来。当然了，一开始大家鞋带都是系好的呦！如果没人改变状态的话，你的好(dou)奇(bi)之心又开始泛滥，想着在这个时刻下，从小伙伴a开始，能否一直将毽子传到小伙伴b那里呢？

要知道：学弟学妹们只能为我们争取T个时刻来玩耍哦。

## 输入格式:

第一行两个整数n和T，表示n个小伙伴参加了游戏，总共有T的时间玩耍。

第二行n个整数，第i个数right[i]，表示第i个小伙伴右边的人是right[i]，保证n个数各不相同。

接下来的T行，每行第一个数op，

若op=1，则接着一个数c，表示第c位同学改变一次状态。

若op=2，则接着两个数a, b，表示询问第a位同学能否将毽子传到第b位同学那里。

## 输出格式:

对于每个op=2，如果可以传到，输出“Yes”，否则输出“No”。

## 样例输入(shuttlecock.in):

```
8 8
5 2 1 6 3 7 8 4
1 3
2 1 5
1 5
2 1 5
1 6
2 4 2
1 6
2 4 7
```

## 样例输出(shuttlecock.out):

```
Yes
No
No
Yes
```

## 数据范围:

30%的数据：  $1 \leq n \leq 100$  ,  $1 \leq T \leq 100$  ,

另40%的数据：  $1 \leq n \leq 10^5$  ,  $1 \leq T \leq 10^5$  , 且所有人构成一个1、2、3、4、....、n的

大圈。即  $\text{right}[1]=2, \text{right}[2]=3, \text{right}[3]=4, \dots, \text{right}[n-1]=n, \text{right}[n]=1$ 。  
100%的数据:  $1 \leq n \leq 10^5, 1 \leq T \leq 10^5, 1 \leq a, b, c \leq n$