

# 2024 西安市信息学算法编程大赛

## 初中组复赛

时间：2024 年 4 月 27 日 14:00 ~ 17:00

题目名称	选购冰激淋	吃饭	冰激淋促销	聪明的质检员
题目类型	传统型	传统型	传统型	传统型
目录	buy	eat	sale	check
可执行文件名	buy	eat	sale	check
输入文件名	buy.in	eat.in	sale.in	check.in
输出文件名	buy.out	eat.out	sale.out	check.out
每个测试点时限	1.0 秒	1.0 秒	3.0 秒	1.0 秒
内存限制	128 MiB	512 MiB	256 MiB	512 MiB
测试点题目	10	10	10	20
测试点是否等分	是	是	是	是

提交源文件程序名

buy.cpp	eat.cpp	sale.cpp	check.cpp
---------	---------	----------	-----------

编译选项

-O2 -std=c++14
----------------

### 注意事项（请仔细阅读）

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C/C++ 中函数 main() 的返回值类型必须是 int，程序正常结束时的返回值必须是 0。
3. 提交的程序代码文件的放置位置请参考考场要求。
4. 因违反以上三点而出现的错误或问题，申诉时一律不予受理。
5. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
6. 选手提交的程序源文件必须不大于 100 KB。
7. 程序可使用的栈空间内存限制与题目的内存限制一致。

## 选购冰激淋 (buy)

### 【题目描述】

你正在经营一家冰淇淋店, 店里所售卖的冰淇淋主料与添加进冰淇淋的配料可以由顾客任选。

目前准备了  $n$  种冰淇淋的主料与配料, 其中有可供选择的主料有 3 种, 其余的均为配料。1 个冰淇淋必须选择 1 种主料, 并且只能选择一种主料, 添加的配料的选择不受限制, 因此可以随意互相配置。配料或者主料不同的冰淇淋被认为是不同的口味。(请注意 int 数据类型能存储的数值范围, 调配冰淇淋至少要放入 1 种主料)

现在你要求算出可以调配出的不同口味冰淇淋的数量总和。

### 【输入格式】

从文件 `buy.in` 中读入数据。

一个整数  $n$ , 意义见题目描述。

### 【输出格式】

输出到文件 `buy.out` 中。

一个整数  $x$ , 表示你可以调配出的不同口味冰淇淋的数量总和。

### 【样例 1 输入】

3

### 【样例 1 输出】

3

### 【样例 2 输入】

4

### 【样例 2 输出】

6

### 【样例 1 解释】

只有 3 种主料, 因此只能配出 3 种冰淇淋: 只用主料 1、只用主料 2、只用主料 3。

### 【样例 2 解释】

你有 3 种主料, 还有 1 种配料, 因此能配出 6 种冰淇淋: 只用主料 1、只用主料 2、只用主料 3、用主料 1 和配料 4、用主料 2 和配料 4、或者用主料 3 和配料 4。

### 【数据范围】

对于 10% 的数据,  $n \leq 5$ 。

对于 100% 的数据,  $n \leq 50$ 。

## 吃饭 (eat)

### 【题目描述】

小明饿了, 所以小明要去吃饭。桌子上摆了  $n$  道菜, 第  $i$  道菜的编号是  $i$ , 美味程度是  $a_i$ 。小明可以选择从任意一道菜开始吃 (也可以不吃离场)。

由于餐厅的额外要求: 假设小明吃的上一道菜的编号是  $x$ , 那么小明吃的下一道菜的编号必须是  $x$  的倍数。

问: 小明吃到的菜的美味程度之和最多是多少。

### 【输入格式】

从文件 `eat.in` 中读入数据。

总共两行。

第一行输入  $n$ 。

第二行输入  $n$  个整数, 表示  $a_1, \dots, a_n$ 。

### 【输出格式】

输出到文件 `eat.out` 中。

一个整数, 表示答案。

### 【样例 1 输入】

```
5
1 2 3 4 5
```

### 【样例 1 输出】

```
7
```

### 【样例 2 输入】

```
5
1 -1 4 7 5
```

### 【样例 2 输出】

```
8
```

### 【样例解释 1】

小明先吃 1, 再吃 2, 再吃 4, 可以得到  $1+2+4=7$  的美味程度。

### 【样例解释 2】

小明先吃 1, 再吃 4, 可以得到  $1+7=8$  的美味程度。

**【数据范围】**

对于 20%的数据,  $n \leq 20$ 。

对于 30%的数据,  $n \leq 30$ 。

对于 60%的数据,  $n \leq 10000$ 。

对于 100%的数据,  $n \leq 2 \times 10^5$ ,  $-10^5 \leq a_i \leq 10^5$ 。

特殊数据: 对于额外 20%的数据, 保证  $a_i = i$ 。

## 冰淇淋促销 (sale)

### 【题目描述】

最近你发现冰淇淋店的销量有所下滑,作为店主的你决定设计一个促销活动来吸引顾客、提高营业额。你想起来了上数学课上学习过的素数概念,这启发你设计一个有关素数的促销活动。

活动开启时店门口会摆放  $n$  张写有不同素数  $p_1, p_2, \dots, p_n$  的广告牌, 每位到店顾客会得到一张印有自己到店顺序  $k$  (例如第一位顾客  $k$  取值为 1, 第二位顾客  $k$  取值为 2, ...) 的兑奖票。所有顾客中仅会产生一位幸运儿可以得到免费的冰淇淋, 他\她所持有的兑奖票  $k$  值需要满足以下的条件:

条件 1:  $k$  的所有质因子均为  $n$  张广告牌上所给出的素数  $p_1, p_2, \dots, p_n$  中之一的数;

条件 2: 在所有满足条件 1 的数中从小到大序列中第  $x$  个数字,  $x$  由店主给出。

满足条件 1 和条件 2 的答案保证小于  $10^{18}$ 。

### 【输入格式】

从文件 `sale.in` 中读入数据。

第一行一个整数  $n$ ;

第二行有  $n$  个用空格隔开的整数, 为  $p_1, p_2, \dots, p_n$ ;

第三行一个整数  $x$ ;

具体意义见题面描述相同

### 【输出格式】

输出到文件 `sale.out` 中。

一个整数, 即幸运儿所持的兑奖票编号  $k$ 。

### 【样例 1 输入】

```
3
2 3 5
7
```

### 【样例 1 输出】

```
8
```

### 【样例解释 1】

满足条件 1 的兑奖票号有 1、2、3、4、5、6、8, 第 7 个满足条件 1 的数字为 8 即答案为 8, 注意 1 也是满足条件 1 的数字。

### 【数据范围】

对于 50% 的数据,  $1 \leq n \leq 8$ 。

对于 100% 的数据,  $1 \leq n \leq 16, 1 < p_i \leq 97, 1 \leq x \leq 10^9$ 。

答案保证小于  $10^{18}$ 。

## 聪明的质检员 (check)

### 【题目描述】

小明是一名质检员，今天要去一个工厂去检查产品质量。

这个工厂提供了  $n$  个样品，编号为  $1, 2, \dots, n$ ，其中，有一个样品是不合格产品，这个不合格产品会等概率的在这  $n$  个样品中产生。

小明有一台检测设备，每次检测，小明可以选择一个样品，然后放入检测设备，有以下两种情况：

(1) 如果小明放进去的样品刚好就是那个不合格产品，则设备会直接识别出结果，小明的任务就完成了。

(2) 如果小明放进去的样品是合格品，则设备有  $p$  的概率，正确地提示小明不合格产品的编号比这个样品大还是小，有  $1-p$  的概率，错误地提示小明不合格样品的编号比这个样品大还是小。

举个例子，比如现在一共有 10 个样品， $p=0.7$ ，不合格样品实际上是在 3，那么，如果小明把第 5 个样品投入进去，则机器有 0.7 的概率提示小明不合格样品在  $[1, 4]$  中，0.3 的概率提示小明不合格样品在  $[6, 10]$  中。

又比如，当前不合格样品只剩两种可能，则无论小明用哪一个样品来检测，机器都只能返回正确，或是样品在另一个里，就不存在  $1-p$  的概率瞎说了。

小明想找到不合格样品的编号，于是他决定，执行下述算法：

假设当前不合格样品所在区间是  $[l, r]$ ，首先用机器检测  $\lfloor \frac{l+r}{2} \rfloor$  这个编号的样品  $k$  次（保证  $k$  是奇数）。

如果这个样品恰好是不合格品，则完成任务。

否则，认可机器回答较多的那一边，然后继续执行上述算法。

举个例子，比如小明询问了样品 5 一共 7 次，机器有 4 次回答了左边，3 次回答了右边，则小明就认为不合格品在 5 的左边。

现在，我们知道  $n, p, k$ ，请帮助小明算一算，他的这个算法有多少概率能找到不合格品。

### 【输入格式】

从文件 `check.in` 中读入数据。

第一行输入三个数。其中是类型的实数。

### 【输出格式】

输出到文件 `check.out` 中。

输出一个实数表示答案，保留四位小数。

### 【样例 1 输入】

```
5 1.00 1
```

### 【样例 1 输出】

```
1.0000
```

**【样例 2 输入】**

3 0.8 1
---------

**【样例 2 输出】**

0.8667
--------

**【样例 3 输入】**

10 0.8 3
----------

**【样例 3 输出】**

0.8412
--------

**【样例解释 1】**

机器不会说谎，小明自然永远不会算错。

**【样例解释 2】**

如果不合格品在 2，则小明第一次就会找到不合格品。

如果不合格品在 1 或者 3，则小明第一次询问 2 的时候，有 0.8 的概率被引导到正确的位置，0.2 的概率引导到错误的位置。

总概率是  $0.8 \times \frac{2}{3} + 1 \times \frac{1}{3} = 0.866667$ 。

**【数据范围】**

测试点 1:  $p=1$ 。

测试点 2-4:  $n \leq 10, p = 0.8$ 。

测试点 5-6:  $k=1$ 。

测试点 7-10:  $n \leq 1000$ 。

测试点 11-16:  $n \leq 10^5$ 。

测试点 1-20:  $1 \leq n \leq 10^{16}, 1 \leq k \leq 11, 0.5 \leq p \leq 1$ 。保证  $k$  是奇数。