

“科大国创杯”
2025 年安徽省青少年信息学科普日活动
小学组试题

时间：2025 年 4 月 19 日 09:00 ~ 12:00

题目名称	饼干	方差	正方形划分	改写
题目类型	传统型	传统型	传统型	传统型
目录	cookie	variance	square	rewrite
可执行文件名	cookie	variance	square	rewrite
输入文件名	cookie.in	variance.in	square.in	rewrite.in
输出文件名	cookie.out	variance.out	square.out	rewrite.out
测试点时限	1.0 秒	1.0 秒	1.0 秒	1.0 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
数据组数	10	10	10	20

提交源程序文件名

对 C++ 语言	cookie.cpp	variance.cpp	square.cpp	rewrite.cpp
----------	------------	--------------	------------	-------------

编译选项

对 C++ 语言	-O2 -std=c++14
----------	----------------

注意事项与提醒：

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() 的返回类型必须是 int, 程序正常结束时的返回值必须是 0。
3. 选手需要要在桌面上建立以选手参赛号为名的目录, 并由选手为每道题再单独建立一个子目录, 子目录名与对应试题的英文名相同。选手提交的每道试题的源程序必须存放在相应的子目录下。

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。如果现场有不同要求, 以现场为准。
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. 程序可使用的栈内存空间限制与题目的内存限制一致。
7. 只提供 Linux 格式的附加样例文件。
8. 评测在当前最新的 NOI Linux 下进行, 各语言的编译器版本以其为准。

1 饼干

(cookie.cpp)

1.1 题目描述

小可可有一个 $1 \times n$ 的长方形盘子，现在他想在里面放一些 $1 \times m$ 的饼干，饼干不能重叠，他想知道他最多能放多少块饼干？

1.2 输入格式

从 cookie.in 输入。

一行两个整数 n, m ，如题面定义。

1.3 输出格式

向 cookie.out 输出。

一行一个整数，表示最多能放的饼干的数量。

1.4 输入输出样例 1

1.4.1 输入样例

19 6

1.4.2 输出样例

3

1.5 输入输出样例 2/3

详见选手文件夹下的 cookie/cookie*.in/ans 文件。

1.6 约定和数据范围

数据点 1 ~ 3, $1 \leq n \leq 10^9, m = 1$ 。

数据点 4 ~ 5, $1 \leq n, m \leq 10^9, n = m$ 。

数据点 6 ~ 10, $1 \leq n, m \leq 10^9$ 。

2 方差

(variance.cpp)

2.1 题目描述

小可可最近学习了方差的定义。

现在有一个长度为 n 的序列 a ，小可可希望你求出这个序列的方差。形式化的，即让你求 $\sigma = \frac{(a_1 - \bar{a})^2 + (a_2 - \bar{a})^2 + \dots + (a_n - \bar{a})^2}{n}$ ，其中 $\bar{a}$ 表示 a 序列的平均数，即为 $\frac{a_1 + a_2 + \dots + a_n}{n}$ ，保证运算过程中所有结果为整数。

2.2 输入格式

从 variance.in 输入。

第一行一个整数 n ，表示序列长度。

第二行 n 个整数，第 i 个整数表示 a_i 。

2.3 输出格式

向 variance.out 输出。

一行一个整数，表示 a 序列的方差。

2.4 输入输出样例 1

2.4.1 输入样例

```
3
3 3 6
```

2.4.2 输出样例

```
2
```

2.5 输入输出样例 2/3

详见选手文件夹下的 variance/variance*.in/ans 文件。

2.6 约定和数据范围

数据点 1, $n = 1, 1 \leq a_i \leq 100$ 。

数据点 2 ~ 5, $1 \leq n \leq 2 \times 10^3, 1 \leq a_i \leq 2 \times 10^5$ 。

数据点 6 ~ 10, $1 \leq n \leq 10^5, 1 \leq a_i \leq 10^7$ 。

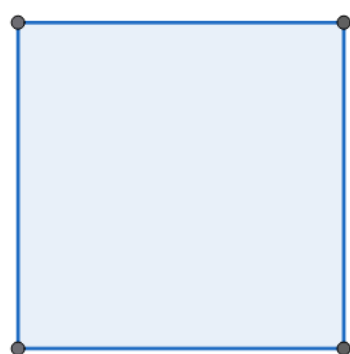
3 正方形划分

(square.cpp)

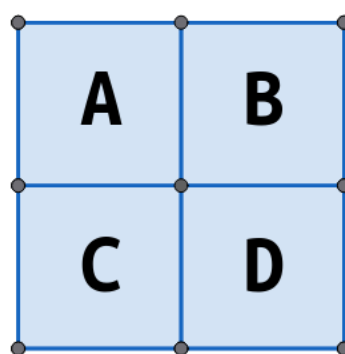
3.1 题目描述

小可可发明了一种新的正方形划分方法。

首先我们有一个正方形，我们称该图形为第 0 轮的图形。我们将这个正方形均匀划分成 4 个部分——左上、右上、左下和右下，分别将其编号为 ABCD，称新的图形为第 1 轮的图形。

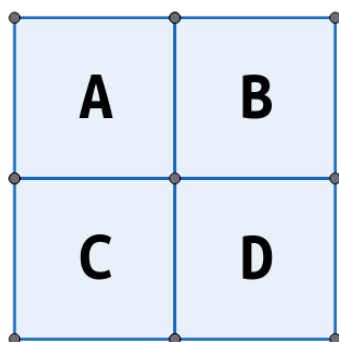


第0轮

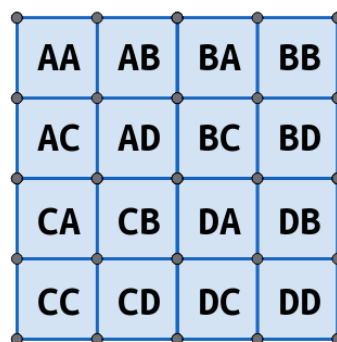


第1轮

我们再将第一个图形所划分的每一个部分 ABCD，分别再分成 4 个部分，称新的图形为第 2 轮的图形。对于该图形的每个部分的命名方式为该部分所属第 1 轮的部分的编号 + ABCD。如：原来第 1 轮的 A 部分的右上部分的编号为 AB。



第1轮



第2轮

第 3 轮及以上的图形以此类推。

这样每个方格都有一个编号，同时也有一个位置，即第几行（从上往下数）第几列（从左往右数）。现在小可想实现编号与其位置的相互转化，请你写一份程序帮帮他。

3.2 输入格式

从 square.in 输入。

本题有多组测试数据，输入第一行一个整数 T 表示测试数据组数，接下来输入每组数据。

对于每组测试数据共一行，有以下两种可能：

0 n x y 表示该测试数据要求出第 n 轮的图形中第 x 行第 y 列的格子的编号。

1 str 表示该测试数据要求出编号为 str 的格子的位置。

如果仍然对输入方式不够清楚，可以去观察输入输出样例和样例解释。

3.3 输出格式

向 square.out 输出。

对于每组测试数据：

如果输入的测试数据为 0 n x y 的形式，输出一行一个字符串 str 表示该格子编号。

如果输入的测试数据为 1 str 的形式，输出一行三个整数 n, x, y 表示该格子位于第 n 轮图形的第 x 行第 y 列。

3.4 输入输出样例 1

3.4.1 输入样例

```
3
0 1 2 1
1 AB
0 2 3 4
```

3.4.2 输出样例

```
C
2 1 2
DB
```

3.4.3 样例解释

第一行一个整数 3 表示我们一共有三组测试数据。

第一组测试数据 0 1 2 1，表示我们要将第 1 轮图形的第 2 行第 1 列的位置转化为编号，根据题意描述中的图可知编号为 C。

第二组测试数据 1 AB, 表示我们要求出编号为 AB 的格子位置, 根据题意描述中的图可知它在第 2 轮第 1 行第 2 列。

第三组测试数据 0 2 3 4, 表示我们要将第 2 轮图形的第 3 行第 4 列的位置转化为编号, 根据题意描述中的图可知编号为 DB。

3.5 输入输出样例 2/3/4/5/6/7

详见选手文件夹下的 square/square*.in/ans 文件。

3.6 约定和数据范围

数据点 1, $1 \leq T \leq 10, 1 \leq n \leq 2$ 。

数据点 2, 3, $1 \leq T \leq 10, 1 \leq n \leq 10$ 。

数据点 4, 5, $1 \leq T \leq 5 \times 10^4, n = 10$, 即保证所有图形均为第 10 轮图形。

数据点 6, $1 \leq T \leq 5 \times 10^4, 1 \leq n \leq 30$, 询问仅形如 0 n x y。

数据点 7, $1 \leq T \leq 5 \times 10^4, 1 \leq n \leq 30$, 询问仅形如 1 str。

数据点 8 ~ 10, $1 \leq T \leq 5 \times 10^4, 1 \leq n \leq 30$ 。

4 改写

(rewrite.cpp)

4.1 题目描述

小可可在学习字符串算法！

一个长度为 m 的字符串 r 是回文的，当且仅当 $r_i = r_{m+1-i}$ 对所有 $1 \leq i \leq m$ 均成立。例如 `aaabaaa`, `abba` 都是回文串，但 `aaabaa` 不是回文串。

给定一个字符串 s ，把 s 分成若干个非空子段，使得每一个子段都不是回文的，同时最大化划分出的子段数目，请你输出最大划分数，无解则输出 -1 。

子段的定义为：一个字符串保留任意连续字符后形成的字符串。

由于字符串 s 可能很长，我们将会按照 c, len 的形式给出整个字符串，具体含义见输入格式。

4.2 输入格式

从 `rewrite.in` 输入。

第一行一个整数 T 表示数据组数。

对于每组数据，第一行输入一个数 n 表示极长连续相同字母段的数量。

接下来 n 行中，第 i 行包括一个 a 到 z 之间的小写字母 c_i 和一个整数 len_i ，分别表示该段的数字以及长度，保证对于所有大于 1 的 i ，均满足 $c_{i-1} \neq c_i$ 。

4.3 输出格式

向 `rewrite.out` 输出。

输出共 T 行，每行输出一个整数，表示最大的划分段数，若无解则输出 -1 。

4.4 输入输出样例 1

4.4.1 输入样例

```
5
2
b 1
a 1
1
b 4
7
a 2
b 2
a 1
```

b 2
a 1
b 1
a 1
5
a 2
b 1
a 2
c 2
a 2
3
a 1
b 1
a 1

4.4.2 输出样例

1
-1
4
3
-1

4.4.3 样例解释

对于第一组数据，序列为 `ba`，且只存在 `ba` 这一种划分方案，因此答案为 1。

对于第二组数据，序列为 `bbbb`，显然没有合法方案。

对于第三组数据，序列为 `aabbabbaba`，存在一种划分出四段的方案：`aabb,ab,ba,ba`，可以证明没有答案更优的划分方式。

对于第四组数据，序列为 `aabaaccaa`，存在一种划分出三段的方案：`aaba,ac,caa`，可以证明没有答案更优的划分方式。

对于第五组数据，序列为 `aba`，容易发现无论怎么划分，都至少有一个回文串，所以无解。

4.5 输入输出样例 2/3/4

详见选手文件夹下的 `rewrite/rewrite*.in/ans` 文件。

4.6 约定和数据范围

数据点 1, $T = 10, 1 \leq n \leq 3, 1 \leq len_i \leq 2$ 。

数据点 2, $T = 10, 1 \leq n \leq 3, 1 \leq len_i \leq 10^9$ 。

数据点 3, 4, $T = 10, 1 \leq n \leq 150, 1 \leq len_i \leq 2$ 。

数据点 5, 6, $T = 10, 1 \leq n \leq 150, 1 \leq len_i \leq 10^9$ 。

数据点 7 ~ 9, $T = 10, 1 \leq n \leq 2.5 \times 10^3, 1 \leq len_i \leq 2$ 。

数据点 10 ~ 12, $T = 10, 1 \leq n \leq 2.5 \times 10^3, 1 \leq len_i \leq 10^9$ 。

数据点 13 ~ 16, $T = 10, 1 \leq n \leq 10^5, 1 \leq len_i \leq 2$ 。

数据点 17 ~ 20, $T = 10, 1 \leq n \leq 10^5, 1 \leq len_i \leq 10^9$ 。