

第四届（2025）青少年编程挑战

精英赛预备组试卷

考试时间：2025 年 10 月 12 日

题目名称	买大米	自除数计数	字符串区间统计	填海造岛
题目类型	传统型	传统型	传统型	传统型
目录	rice	divisior	string	land
可执行文件名	rice	divisior	string	land
输入文件名	rice.in	divisior.in	string.in	land.in
输出文件名	rice.out	divisior.out	string.out	land.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	1.0 秒
内存限制	512MB	512MB	512MB	512MB
子任务数目	10	10	10	10
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	rice.cpp	divisior.cpp	string.cpp	land.cpp
对于 C 语言	rice.c	divisior.c	string.c	land.c

编译选项

对于 C++ 语言	-O2 -lm
对于 C 语言	-O2 -lm

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文件提交格式的具体要求。
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. 全省统一评测时采用的机器配置为：Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1.99 GHz 16G 内存。上述时限以此配置为准。
7. 只提供 Linux 格式附加样例文件。
8. 评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以此为准。

买大米 (rice)

【题目描述】

小李在家做饭，发现没有米了，他打算拿着 n 元钱去超市买大米。假如超市的大米 5.8 元/ kg ，小李购买了 m kg 大米。如果可以购买，输出购买大米后剩下多少钱；如果不可以购买，输出差多少钱？（答案保留1位小数）

【输入格式】

从文件 **rice.in** 中读入数据。

一行两个正整数 n 、 m ，表示小李拿着 n 元钱去买 m kg 大米。

【输出格式】

输出到文件 **rice.out** 中。

输出一行一个数。如果可以购买，输出购买大米后剩下多少钱；如果不可以购买，输出差多少钱？

【样例 1 输入】

```
1 50 5
```

【样例 1 输出】

```
1 21.0
```

【样例 1 解释】

要购买 $5kg$ 大米，需要 $5 \times 5.8 = 29$ 元，买完大米后剩下 21 元，输出 21.0 。

【样例 2 输入】

```
1 50 12
```

【样例 2 输出】

```
1 19.6
```

【样例 2 解释】

要购买 $12kg$ 大米，需要 $12 \times 5.8 = 69.6$ 元，不可以购买大米差 19.6 元，输出 19.6。

【数据范围】

对于所有测试数据保证： $1 \leq n, m \leq 10^9$ 。

自除数计数 (divisor)

【题目描述】

朋朋最近了解了“自除数”的概念：如果一个数不含数字 0，且能被它包含的每一位数字整除，那么这个数就是一个自除数。例如，128 是自除数，因为 128 不含 0 且 $128 \div 1 = 128$ 、 $128 \div 2 = 64$ 、 $128 \div 8 = 16$ ，所有除法均能除尽。

现在朋朋想计算一个给定区间 $[left, right]$ 内有多少个自除数。由于区间范围可能很大，逐个判断每个数字会非常耗时。请你编写一个程序，高效地帮助朋朋计算出区间内自除数的个数。

【输入格式】

从文件 **divisor.in** 中读入数据。

一行两个正整数，表示区间 $left$ 和区间 $right$ 。

【输出格式】

输出到文件 **divisor.out** 中。

一行一个正整数，表示区间内的自除数有几个。

【样例 1 输入】

```
1 1 22
```

【样例 1 输出】

```
1 13
```

【样例 1 解释】

在 $[1, 22]$ 区间内自除数有 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 15, 22 共计 13 个故输出 13。

【样例 2 输入】

```
1 15 50
```

【样例 2 输出】

1	7
---	---

【样例 2 解释】

在 $[15,50]$ 区间内自除数有15,22,24,33,36,44,48共计 7 个故输出 7。

【数据范围】

对于所有测试数据保证： $1 \leq left < right \leq 10^7, right - left \leq 10^7$ 。

字符串区间统计（string）

【题目背景】

在文本处理中，经常需要统计某一段文本中特定字符或子串的出现频率。为了提高查询效率，可以预先计算前缀和数组。

【题目描述】

给定一个由小写字母组成的字符串 S ，然后进行 q 次查询。每次查询给出两个整数 L 和 R （ $1 \leq L \leq R \leq n$ ，其中 n 为字符串 S 的长度）以及一个小写字母 c ，询问在字符串 S 的子串 $S[L..R]$ 中，字母 c 出现的次数。

【输入格式】

从文件 **string.in** 中读入数据。

第一行一个字符串 S 。

第二行一个整数 q ，表示查询次数。

接下来 q 行，每行首先两个整数 L 和 R ，然后一个字符 c （保证为小写字母），表示查询子串 $S[L, R]$ 中字符 c 的出现次数。

【输出格式】

输出到文件 **string.out** 中。

对于每个查询，输出一行一个整数，表示该字符在子串中出现的次数。

【样例 1 输入】

```
1 aabac
2 3
3 1 3 a
4 2 4 b
5 1 5 c
```

【样例 1 输出】

1	2
2	1
3	1

【样例 1 解释】

在字符串 `aabac` 中，对于查询 1:在 $[1,3]$ 区间内 `a` 出现了 2 次，即 `aab`；对于查询 2: 在 $[2,4]$ 区间内 `b` 出现了 1 次，即 `aba`；对于查询 3: 在 $[1,5]$ 区间内 `c` 出现了 1 次，即 `aabac`。

【样例 2】

见选手目录下 `string/ string2.in` 和 `string / string2.ans`。

【样例 3】

见选手目录下 `string/ string3.in` 和 `string / string3.ans`。

【数据范围】

对于所有测试数据保证：字符串长度 $length$ 不超过 10^5 ,查询次数 q 均不超过 10^7 。

填海造岛 (land)

【题目描述】

自 2013 年起，为守护国家主权和海洋权益，中国在南海开展了大规模的岛礁填海工程。经过多年建设，已在南海成功填建了美济岛、永暑岛、渚碧岛等 8 个重要岛屿。十年前，这些岛礁在涨潮时几乎被海水淹没，驻守官兵只能居住在狭小的高脚屋中；如今，这些地方已发展成为功能完善的现代化海上新城，配备了机场、医院、农场和学校等设施。

现在正在规划填海建造一个新的人工岛，给定一个 $n \times n$ 的二进制矩阵来表示海域，其中 1 表示陆地，0 表示海水。最多只能将一处海水 (0) 变为陆地 (1)，请计算填海后能够形成的最大陆地面积。

注意：陆地的面积指的是相连的 1 所组成的区域大小。相连规则采用四连通（即上下左右四个方向相邻的 1 视为相连）。

【输入格式】

从文件 **land.in** 中读入数据。

第一行一行一个正整数 n ，表示区域的大小。

接下来的 n 行，一行 n 个数表述陆地和海水。

【输出格式】

输出到文件 **land.out** 中。

一行一个正整数，表示 $n \times n$ 的区域内陆地的最大面积。

【样例 1 输入】

```
1 2
2 1 0
3 0 1
```

【样例 1 输出】


```
1 3
```

【样例 1 解释】

如图所示：将左下角的海水（0）变为陆地（1），陆地的最大面积为 3。

1	0
1	1

【样例 2 输入】

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

【样例 2 输出】

```
1 3
```

【样例 2 解释】

如图所示：将第一列中间的海水（0）变为陆地（1），陆地的最大面积为 3

1	0	1
1	0	0
1	0	1

【样例 3 输入】

```
1 5
2 1 0 1 1 0
3 1 0 0 0 1
4 0 0 0 1 1
5 1 0 1 1 0
6 0 0 0 0 0
```

【样例 3 输出】

1

8

【样例 3 解释】

如图所示：将第二行第 4 个海水（0）变为陆地（1），陆地的最大面积为 8

1	0	1	1	0
1	0	0	1	1
0	0	0	1	1
1	0	1	1	0
0	0	0	0	0

【样例 4】

见选手目录下 land/ land4.in 和 land/ land4.ans。

【数据范围】

对于所有测试数据保证： $1 \leq n \leq 1000$ 。