

预备组模拟卷

题目名称	与圆相关的计算	完美正整数	统计单词数	猴群
题目类型	传统型	传统型	传统型	传统型
目录	circle	perfect	words	monkeys
可执行文件名	circle	perfect	words	monkeys
输入文件名	circle.in	perfect.in	words.in	monkeys.in
输出文件名	circle.out	perfect.out	words.out	monkeys.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	128MB	128MB
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	circle.cpp	perfect.cpp	words.cpp	monkeys.cpp
对于 C 语言	circle.c	perfect.c	words.c	monkeys.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。

与圆相关的计算(circle)

【题目描述】

输入圆的半径,求圆的直径(直径=半径 $\times 2$),周长(周长=直径 $\times 3.1415926$),面积(面积=3.1415926 \times 半径 \times 半径),保留小数点后4位。

【输入格式】

从文件 **circle.in** 中读入数据。

输入包含一个实数 $r(0 < r \leq 10,000)$,表示圆的半径。

【输出格式】

输出到文件 **circle.out** 中。

输出一行,包含三个数,分别表示圆的直径、周长、面积,数与数之间以一个空格分开,每个数保留小数点后4位。

【样例 1 输入】

1 2.5

【样例 1 输出】

1 5.0000 15.7080 19.6350

【样例 2 输入】

1 100.5

【样例 2 输出】

1 201.0000 631.4601 31730.8707

完美正整数(perfect)

【题目描述】

小 AI 最近在研究数学问题，他自己发明了他心中的完美正整数。符合下列条件的正整数 N 在小 AI 心中即为完美正整数：

1. N 为一个完全平方数；

2. N 的各位数位和为素数； 也就是说，25 这个数字，是小 AI 心中的完美正整数。

因为 $25=5\times 5$ ，符合要求 1，并且 $2+5=7$ ，7 为素数，符合要求 2。现在小 AI 遇到了一个问题，他希望找出 L 到 R 的完美正整数是哪几个？请你写一个程序帮助他。

【输入格式】

从文件 `perfect.in` 中读入数据。

一行，输入两个正整数 L, R 。

【输出格式】

输出到文件 `perfect.out` 中。

输出 $[L, R]$ 中的完美正整数（从小到大）。若不存在优秀正整数，则输出 0。

【样例 1 输入】

```
1 1 30
```

【样例 1 输出】

```
1 16 25
```

【样例 1 解释】

$[1,30]$ 这个区间中，16 和 25 符合完美正整数的定义，故输出为 16 和 25。

测试点	L, R≤
1~2	10^3
3~5	10^5
6~7	10^7
8~10	10^{12}

统计单词数(words)

【题目描述】

一般的文本编辑器都有查找单词的功能,该功能可以快速定位特定单词在文章中的位置,有的还能统计出特定单词在文章中出现的次数。

现在,请你编程实现这一功能,具体要求是:给定一个单词,请你输出它在给定的文章中出现的次数和第一次出现的位置。注意:匹配单词时,不区分大小写,但要求完全匹配,即给定单词必须与文章中的某一独立单词在不区分大小写的情况下完全相同(参见样例1),如果给定单词仅是文章中某一单词的一部分则不算匹配(参见样例2)。

【输入格式】

从文件 `words.in` 中读入数据。

每组输入数据的第一行为一个字符串,其中只含字母,表示给定单词;

第二行为一个字符串,其中只可能包含字母和空格,表示给定的文章。(1≤单词长度≤10, 1≤文章长度≤1,000,000)

【输出格式】

输出到文件 `words.out` 中。

每组输出只有一行,如果在文章中找到给定单词则输出两个整数,两个整数之间用一个空格隔开,分别是单词在文章中出现的次数和第一次出现的位置(即在文章中第一次出现时,单词首字母在文章中的位置,位置从0开始); 如果单词在文章中没有出现,则直接输出一个整数-1。

【样例1输入】

```
1 To
2 to be or not to be is a question
```

【样例1输出】

1 2 0

【样例 2 输入】

1 to

2 Did the Ottoman Empire lose its power at that time

【样例 2 输出】

1 -1

【样例 解释】

样例一：

输出结果表示给定的单词 To 在文章中出现两次，第一次出现的位置为 0。

样例二：

表示给定的单词 to 在文章中没有出现，输出整数-1。

猴群 (monkeys)

【题目描述】

给出一个由数字 0~9 组成的矩形，其中数字 0 代表树，1~9 代表猴子，凡是由 0 或矩形边围起来的区域表示有一群猴子在这一带。编程求矩形中有多少群猴子。

【输入格式】

从文件 `monkeys.in` 中读入数据。

第一行两个正整数，表示矩形的行数 m 和列数 n ， $1 \leq m, n \leq 100$ ；

下面为一个 $m \times n$ 的数字矩形。

【输出格式】

输出到文件 `monkeys.out` 中。

一行一个数，表示猴群的数目。

【样例 1 输入】

```
1 4 10
2 0234500067
3 1034560500
4 2045600671
5 0000000089
```

【样例 1 输出】

```
1 4
```