

2026 第五届昆明青少年人工智能综合实践大赛

Python 编程决赛（提高组）

考试时间：2026 年 4 月 25 日 14: 00-18: 00

题目名称	湖泊大小	因子之和	快递配送	墓穴探险
题目类型	传统型	传统型	传统型	传统型
目录	island	factor	express	tomb
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	1.0 秒
内存限制	256MB	256MB	256MB	512MB
子任务数目	10	10	20	20
测试点是否等分	是	是	是	是

提交源程序文件名

对于 Python 语言	island.py	factor.py	express.py	tomb.py
--------------	-----------	-----------	------------	---------

编译选项

对于 Python 语言	
--------------	--

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. Python 中函数 main() 的返回值类型必须是。
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。

湖泊大小 (island)

【题目描述】

用一个 n 行 m 列的区域之内有若干个湖泊，一个湖泊由上下左右相邻的字符 ‘0’ 网格组成。除了湖泊的网格外，都是地面网格，地面网格都是用字符 ‘1’ 表示。

求网格数最多的湖泊的大小、网格数最少的湖泊的大小。

【输入格式】

从文件 **island.in** 中读入数据。

第一行 2 个整数 n 、 m ，表示网格的行数与列数。

接着 n 行，每行 m 个字符，其中字符 0 表示湖泊网格、字符 1 表示地面网格。

【输出格式】

输出到文件 **island.out** 中。

输出仅 2 个整数，为网格数最多的湖泊大小、网格数最少的湖泊的大小。

【样例 1 输入】

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

【样例 1 输出】

```
1 1 1
```

【样例 1 解释】

二维网格情况如下图：

湖	地	地	地
地	地	地	地
地	地	地	地
地	地	地	湖

湖有两个，大小均为 1，那么最大和最小的湖均为 1。

【样例 2 输入】

```

1 5 5
2 00000
3 01110
4 01010
5 01110
6 00000

```

【样例 2 输出】

```

1 12 1

```

【样例解释】

二维网格如下图。

湖1	湖1	湖1	湖1	湖1
湖1	地	地	地	湖1
湖1	地	湖2	地	湖1
湖1	地	地	地	湖1
湖1	湖1	湖1	湖1	湖1

湖有两个，其中一个大小为 12，另外一个大小为 1。

【数据范围】

对于所有测试数据保证： $1 \leq n, m \leq 2000$ 。保证至少有一个湖泊。

测试点	$n, m \leq$
1~2	20
3~4	100
5~6	200
7~8	1000
9~10	2000

因子之和 (factor)

【题目描述】

求 $1\sim n$ 的每个正整数的所有因子之和。

$$\sum_{1 \leq i \leq n} \sum_{d|i} d$$

【输入格式】

从文件 **factor.in** 中读入数据。

输入仅一个数字 n 。

【输出格式】

输出到文件 **factor.out** 中。

输出仅一个数字，即求 $1\sim n$ 的每个正整数的所有因子之和。

【样例 1 输入】

```
1 5
```

【样例 1 输出】

```
1 21
```

【样例 1 解释】

1 的因子只有 1。

2 的因子有 1、2。

3 的因子有 1、3。

4 的因子有 1、2、4。

5 的因子有 1、5。

总和为 $1+1+2+1+3+1+2+4+1+5=21$ 。

【数据范围】

对于所有测试数据保证： $1 \leq n \leq 5 \times 10^6$

测试点	$n \leq$
1~2	10
3~4	10^3
5~6	10^5
7~10	5×10^6

快递配送 (express)

【题目描述】

小张是一名快递员，他需要从配送站出发，将一份紧急快件送到城市另一端的一栋写字楼。

城市被划分成了 $n \times m$ 的网格区域。每个格子要么是普通道路，要么是施工围挡（无法通行），要么是快充站。

小张的电瓶车电量的上限为 k ，每移动一格消耗 1 单位电量（耗时 1 分钟）。如果电量耗尽，车辆就会抛锚，无法继续行驶。起始时，电瓶车的电量是满的。

地图用 $n \times m$ 的网格表示：

- '.'：普通道路，可以通行。
- '#'：施工围挡，无法通行。
- 数字 '1' ~ '9'：快充站。进入该格子后，可以立即给电瓶车充电 x 单位电量（ x 为该数字），但电量不能超过上限 k 。每次经过都可以再次充电。

起点（配送站）和终点（写字楼）保证是普通道路 '.' 网格。

小张想知道：在保证不中途抛锚的前提下，从起点到终点的最短时间（分钟数）是多少？

【输入格式】

从文件 `express.in` 中读入数据。

输入的第一行包括 7 个整数 n, m, sx, sy, ex, ey, k ，分别表示地图行数和列数、起点（配送站）、终点（写字楼）、电量上限。

接下来 n 行，每行 m 个字符，表示矩阵地图每个格子的信息。

【输出格式】

输出到文件 `express.out` 中。

输出仅 1 个数字，如果能够到达，则输出最快到达的时间；如果永远不可能到达，输出 -1。

【样例 1 输入】

```
1 3 10 2 2 2 8 5
2 #####5#####
3 #.....#
4 #####
```

【样例 1 输出】

```
1 8
```

【样例 1 解释】

在样例中，起点为 (2,2)，终点为 (2,8)，唯一的方案为从起点一直走到 (1,6)，充电使得电量变为 5 后，再直接前往终点，一共需要 8 步。

【样例 2】

见选手目录下 `treasure / treasure2.in` 和 `treasure / treasure3.ans`。

【样例 3】

见选手目录下 `treasure / treasure3.in` 和 `treasure / treasure3.ans`。

【样例 4】

见选手目录下 `treasure / treasure4.in` 和 `treasure / treasure4.ans`。

【数据范围】

对于所有测试数据保证： $1 \leq n, m \leq 1000$ ， $1 \leq k \leq 10$ 。

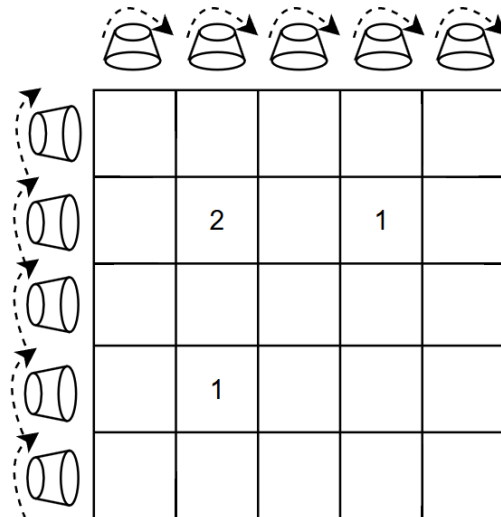
测试点	$n, m \leq$	$k \leq$
1~3	10	10
5	1000	1
6~20	1000	10

墓穴探险 (tomb)

【题目描述】

小向很喜欢玩单机游戏《古墓丽影》，他扮演一个考古学家劳拉在全球各个墓穴探险。其中一个游戏关卡是在一个法老王的墓地探险的过程中，有一个谜题，他试了一整天才解决这个谜题。

谜题是有一个 n 行 m 列的矩阵，每个格子都有一个数字，数字的数值在谜题的一开始都是 0，在矩阵的每行每列都对应有一个可以旋转的按钮，左旋一次这个按钮就可以使得对应行或列矩阵内的所有数字减 1，右旋一次则可以使得对应行或列的矩阵内的所有数字加 1。



游戏关卡内另外有一些 k 条提示线索，每个线索要求 i 行 j 列的格子中最终的数字是 x ，对于没有提示的格子，那些都与谜题无关，可以不关心它的最终数字。

小向花了九牛二虎之力才解开这个谜题，于是他兴奋的向小李吹嘘自己的聪明才智。小李却不屑一顾，认为用最少的操作次数解开才是真正的高手。

于是小李把谜题抽象成若干组数据，需要小向用程序输出最少的左旋/右旋次数。不过小李也很懒，他设置的谜题数据是完全随机的，有时会出现无解的情况，那么对应输出字符串“No”即可。

【输入格式】

从文件 **tomb.in** 中读入数据。每个测试点包含多组测试数据。

输入的第一行是一个正整数 t ，表示测试数据的组数。

每组测试数据的第一行包含三个正整数 n 、 m 、 k 分别表示矩阵的行数、矩阵的列数、谜题的提示数。

接着 k 行，每行三个数字 i 、 j 、 x ，表示一组线索，要求最终 i 行 j 列的格子内的数字是 x 。

【输出格式】

输出到文件 **tomb.out** 中。每组测试数据输出 1 行，如果有解，则输出最少的操作次数，如果无解，输出 “No” 即可。

【样例 1 输入】

```
1 2
2 2 2 4
3 1 1 0
4 1 2 0
5 2 1 2
6 2 2 2
7 2 2 4
8 1 1 0
9 1 2 0
10 2 1 2
11 2 2 1
```

【样例 1 输出】

```
1 2
2 No
```

【样例 1 解释】

输入包含 2 组测试数据。

第一组测试数据谜题的目标如下：

0	0
2	2

操作 2 次第二行的按钮，每次加 1，就可以满足题目条件。明显没有更优的

方案。

第二组测试数据谜题的目标如下：

0	0
2	1

每次操作之后，一定是两个格子的数字发生了加 1 或者减 1，所以矩阵内所有数字之和一定是偶数，但是目标的所有格子的之和是奇数 ($0+0+2+1=3$)，所以必然无解。

【样例 2】

见选手目录下 `tomb / tomb2.in` 和 `tomb / tomb2.ans`。

【样例 3】

见选手目录下 `tomb / tomb3.in` 和 `tomb / tomb3.ans`。

【数据范围】

对于所有测试数据有： $1 \leq t \leq 5, 1 \leq n, m \leq 1000, 1 \leq k \leq n \times m, 1 \leq i \leq n, 1 \leq j \leq m, |x| \leq 10^5$ 。保证两两不同的线索之间的网格均不同。保证每个测试点的 k 之和不会超过 2×10^6 。

测试点	$n, m \leq$	特殊性质
1~3	10	无
4	10	$x = 1$
5~7	50	无
8	50	$n = 1$
9~12	100	无
13	100	$k = 1$
14	500	$n = 1$
15~16	500	$x = i$
17~20	1000	无