

准考证号：SH-B260103

(请选手务必仔细阅读本页内容)

一、题目概况

中文题目名称	投资方案	记单词	新生入学	罗密欧与朱丽叶 3	杂草种子
英文题目与子目录名	invest	letter	shuttle	rome3	seeds
可执行文件名	invest	letter	shuttle	rome3	seeds
输入文件名	invest.in	letter.in	shuttle.in	rome3.in	seeds.in
输出文件名	invest.out	letter.out	shuttle.out	rome3.out	seeds.out
每个测试点时限	1 秒	1 秒	1 秒	1 秒	1 秒
测试点数目	10	10	10	10	10
每个测试点分值	10	10	10	10	10
附加样例文件	无	无	无	无	无
结果比较方式	全文比较，过滤末行后空行，不过滤中间行行末空格				
题目类型	传统	传统	传统	传统	传统
运行内存上限	512M	512M	512M	512M	512M

二、提交源程序文件名

对于 C++ 语言	invest.cpp	letter .cpp	shuttle .cpp	rome3.cpp	seeds.cpp
-----------	------------	-------------	--------------	-----------	-----------

三、编译命令（不包含任何优化开关）

对于 C++ 语言	g++ -o invest - 1m	g++ -o letter letter -1m	g++ -o shuttle shuttle.cpp - 1m	g++ -o letter rome3.cpp - 1m	g++ -o seeds seeds .cpp -1m
-----------	--------------------	--------------------------	---------------------------------	------------------------------	-----------------------------

投资方案

(invest.cpp)

时空限制：1s/512M，测试数据共 10 组

【题目描述】

小菜是一位投资领域的专家，设计了一套先进的量化交易系统。该系统运用尖端算法，能够预测未来 n 天内每天的股票盈利或亏损情况。具体来说，第 i 天的盈利预测为 $x[i]$ 元，若 $x[i]$ 为负值，则表示亏损。基于这些预测，小菜需要制定一个投资策略：选择一段连续的天数进行投资，以实现最大盈利，用 A 来表示这个最大盈利总额。请注意，投资的天数至少为 1 天，不能为 0。同时，有可能最大的盈利 A 是一个负数。

除了计算最大盈利 A 之外，小菜还希望确定达到这一盈利额的不同投资方案的数量，用 B 来表示。值得注意的是，只有当两种投资方案在至少一天上不同时，它们才被视为不同的方案。目前小菜希望在系统中编写新的程序，完成 A 和 B 的高效计算。

【输入格式】

从文件 invest.in 中读入数据。

输入第一行为正整数 n 。第二行为 n 个整数由空格隔开，代表题目中的 $x[1], x[2], \dots, x[n]$ 。

【输出格式】

输出到文件 invest.out 中。

输出共一行，包含两个整数 A 和 B ，由空格隔开。

【输入输出样例 1】

invest.in	invest.out
10 2 -1 -1 3 -1 -2 4 -1 -3 3	4 3

【说明】最多的盈利 $A=4$ ，共 3 种不同的方案：(2 -1 -1 3 -1 -2 4)，(3 -1 -2 4)，(4)。

【输入输出样例 2】

invest.in	invest.out
-----------	------------

5 -6 -6 -6 -6 -6	-6 5
---------------------	------

【说明】 最多的盈利 $A=-6$ ，共 5 种不同的方案：选第 1 天，选第 2 天，选第 3 天，选第 4 天，选第 5 天。

【样例 3】

见目录下的 invest3.in 与 invest3.ans。

【数据规模与约定】

测试点编号	特殊性质
1	所有 $x[i]$ 均为正数
2	所有 $x[i]$ 均为负数
3	所有 $x[i]$ 均相同
4~6	$n \leq 5000$ ， $x[i]$ 的绝对值均不超过 100
7~10	无

保证所有数据： $n \leq 10^5$ ， $x[i]$ 的绝对值均不超过 10^9 。

记单词

(letter.cpp)

时空限制：1s/512M，测试数据共 10 组

【问题描述】

小强最讨厌记单词，尤其是长单词。

小明身为小强的好朋友，告诉了他记单词的秘诀——拆分单词！只要把长单词拆成一小段一小段的小单词，就容易记多啦。为了使小强更容易理解，小明定义一个单词的复杂度为**两两相邻字母 ASCII 码之差的绝对值之和**，例如 bad 的复杂度为 $|b-a|+|a-d|=1+3=4$ 。通过拆分单词，可以让单词的复杂度降低，例如将 bad 拆成 ba 和 d，则复杂度便降成了 1。

可是，让小明没想到的是，单词的数量也会影响小强记单词的效率，假设现在已经有 k 个单词了，如果再多拆除一个单词，小强就会多感受到 k 点复杂度。例如 bad 拆成 b 和 a 和 d，则小强会多感受到 $1+2=3$ 点复杂度。小明已经束手无策了，于是小强向你求助，如何拆解这个单词，可以使总复杂度最低。（所有拆分开单词之间**用空格隔开**，如果有总复杂度相同的方案，则输出**空格的位置最靠前**，且**空格数最少**的方案）

【输入格式】

输入文件 letter.in

输入只有一行，包含一个字符串 s ，表示小强需要记忆的长单词。数据保证字符串中**不会出现空格**，并且字符串**全由小写字母构成**。

【输出格式】

输出文件 letter.out

输出只有一行，一个字符串，表示拆分后的单词。

【输入输出样例 1】

letter.in	letter.out
ace	a ce

【说明】依据题意，拆分后单词的复杂度为 2，小强感受到的复杂度为 1，总复杂度为 $2+1=3$ ，

如果拆分为 a c e，总复杂度也为 $3(0+3=3)$ ，但空格数较多。如果拆分成 ac e，总复杂度也为 3，但空格位置靠后。

【输入输出样例 2】

letter.in	letter.out
apple	a pp le

【输入输出样例 3】

letter.in	letter.out
refrigerator	r ef r ige r a tor

【数据规模与约定】

设单词长度为 n

对于 20% 的数据, $1 \leq n \leq 3$

对于另外 10% 的数据, 单词所有字母均一样

对于 100% 的数据, $1 \leq n \leq 1 \cdot 10^5$

新生入学

(shuttle.cpp)

时空限制：1s/512M，测试数据共 10 组

【问题描述】

每年暑假结束后，大学里就会迎来一批新生入学。全国各地的新生里有不少人会坐火车来到大学所在的城市。学校坐落在一座山顶，火车停靠的站点在山脚下，如果新生自行从火车站前往学校很不方便，容易迷路走失。于是学校做出了以下两个安排：

- 1.在火车站建造宿舍，提供给刚下火车的新生休息，甚至可以住几天。
- 2.安排一辆摆渡车专门负责从火车站将新生送到学校门口。

这辆摆渡车的司机就是人称“魔都车神”的你！为了做好准备，你已经将所有来火车站的学生信息登记在册。共有 n 位新生会来到火车站，编号 1 到 n 。已知 i 号学生会在第 $L[i]$ 天的早晨到达，这位学生愿意等待摆渡车到第 $R[i]$ 天为止。也就是说学生的等待是有忍耐限度的，如果在第 $L[i]$ 天到第 $R[i]$ 天之间没有车能够接这位学生去学校，该学生就会在第 $R[i]+1$ 天一大早选择离开火车站自行前往学校，这种情况太危险了，是所有人都不希望看到的。校长命令你必须开车将所有学生安全送达学校，不可以遗漏任何一人！为此，你可能每天要多次来回接送，当然你车速奇快，无论多少次接送都可以很快完成。另外，每来回一次的油费是 1 百元，每次接送最多带上 k 位新生。请问，你至少要花费几百元油费才能完成任务？

【输入格式】

输入文件 shuttle.in 输入第一行为正整数 n 和 k 。接着 n 行每行包含 2 个正整数依次代表 $L[i]$ 和 $R[i]$ ，保证 $L[i] \leq R[i]$ 。

【输出格式】

输出文件 shuttle.out 输出共 1 行，包含 1 个整数，单位是百元。

【输入输出样例 1】

shuttle.in	shuttle.out
4 2 1 3 2 4 6 7 4 7	2

【说明】有一种合法的两次接送方案：第一次带上 1 号和 2 号学生，发生在第 2 天；第二次带上 3 号和 4 号学生，发生在第 6 天。

【数据规模与约定】

1 号数据: $k=1$

2 号数据: $k=2$

3 号数据: 对于所有 i : $L[i]=R[i]$

4 号数据: 对于所有 i : $R[i]-L[i]$ 均为 10

5-10 号数据: $k \leq n \leq 100000$, $L[i]$ 和 $R[i]$ 均不超过 1000000000

菜鸟编程

罗密欧与朱丽叶 3

(rome3.cpp)

时空限制：1s/512M，测试数据共 10 组

【问题描述】

在一个 $n*m$ 格子的迷宫里，o 代表空地可以行走，#代表墙体不可以通过，R 代表罗密欧，J 代表朱丽叶。罗密欧和朱丽叶每秒钟可以上下左右四个方向走动一格，请问他们相遇需要至少几秒钟？如果无法相遇输出 forever。

【输入格式】

输入文件 romeo3.in

输入第一行为正整数 n 和 m ， $n, m \leq 2000$ 。接着是 $n*m$ 格的字符。**【输出格式】**

输入文件 romeo3.out

一个实数表示答案（精确到 1 位小数），或者输出 forever 表示无解

【输入输出样例 1】

rome3.in	rome3.out
2 2 oJ R#	1.0

【说明】两个人同时往 (1,1) 或 (2,2) 走，即可相遇，需要 1.0 秒。**【输入输出样例 2】**

rome3.in	rome3.out
4 5 ooo#J o#o#o o#o#o R#ooo	6.5

【输入输出样例 3】

rome3.in	rome3.out
2 2 R# #J	forever

【数据规模与约定】

对于 100%的数据 $n, m \leq 2000$ 。

杂草种子

(seeds.cpp)

时空限制: 1s/512M, 测试数据共 10 组

【问题描述】

自然界里的生命来自哪里? 这是个很深奥的问题。最近你突然对这个专题非常感兴趣, 于是你开始研究土壤中的杂草是怎么自发生长的。首先你将一片纯净的农田划分成 $n*m$ 格, 然后你在某些格子里亲手埋入杂草种子, 观察杂草的生长规律: 如果找到 3 个格子 A,B,C 都有杂草种子, 并且格子 B 和 A 是同一行, 格子 C 和 A 是同一列, 那么就会在与 C 同一行且与 B 同一列的第 4 个格子 D 里, 非常神奇地, 从无到有出现杂草种子。也就是说, 若某个平行于农田边框的矩形的三个角上格子已经有种子, 那么第四个角一定会自动长出杂草。

此时此刻, 你的农田里已经有 q 个格子有种子存在, 请问你应该额外添加至少几个格子的种子, 才能使整个农田的 $n*m$ 格都自动被填满杂草?

【输入格式】

输入文件 seeds.in

输入第一行为 3 个正整数 n,m,q 。之后 q 行, 每行 2 个正整数 x,y 代表第 x 行第 y 列已经有种子。保证 x,y 均在 1 到 n 之间。

【输出格式】

输出文件 seeds.out

输出一个整数。

【输入输出样例 1】

seeds.in	seeds.out
2 2 3 1 2 2 2 2 1	0

【输入输出样例 2】

seeds.in	seeds.out
1 5 3	2

1 3	
1 1	
1 5	

【数据规模与约定】

10%数据： $n \leq 2, m \leq 2$

50%数据： $n \leq 20, m \leq 20$

所有数据： 保证 $n, m, q \leq 200000$ 。

葉奧編程