

精英营 day1 习题集

1 基础题（每题 5 分钟完成）

- 1.1 判断素数 (prime.cpp)
- 1.2 最大公约数最小公倍数 (gcd.cpp)
- 1.3 数字翻转 (reverse.cpp)
- 1.4 字符串翻转 (strReverse.cpp)
- 1.5 斐波那契数列 (fibo.cpp)

2 T1 真题（每题 15 分钟）

2.1 NOIP2018 标题统计 (title.cpp)

【题目描述】

凯凯刚写了一篇美妙的作文，请问这篇作文的标题中有多少个字符？ 注意：标题中可能包含大、小写英文字母、数字字符、空格和换行符。统计标题字符数时，空格和换行符不计算在内。

【输入输出格式】

输入格式：

输入文件只有一行，一个字符串。

输出格式：

输出文件只有一行，包含一个整数，即作文标题的字符数（不含空格和换行符）。

【输入输出样例 1】

title.in	title.out
234	3

【输入输出样例 2】

title.in	title.out
ca 45	4

【输入输出样例 1 说明】

标题中共有 3 个字符，这 3 个字符都是数字字符。

【输入输出样例 2 说明】

标题中共有 5 个字符，包括 1 个大写英文字母，1 个小写英文字母和 2 个数字字符，还有 1 个空格。由于空格不计入结果中，故标题的有效字符数为 4 个。

【数据规模与约定】

规定 $|s|$ 表示字符串 s 的长度（即字符串中的字符和空格数）。

对于 40% 的数据， $1 \leq |s| \leq 5$ ，保证输入为数字字符及行末换行符。

对于 80% 的数据， $1 \leq |s| \leq 5$ ，输入只可能包含大、小写英文字母、数字字符及行末换行符。

对于 100% 的数据， $1 \leq |s| \leq 5$ ，输入可能包含大、小写英文字母、数字字符、空格和行末换行符。

2.2 NOIP2017 成绩 (score.cpp)

【问题描述】

牛牛最近学习了 C++ 入门课程，这门课程的总成绩计算方法是：

总成绩 = 作业成绩 \times 20% + 小测成绩 \times 30% + 期末考试成绩 \times 50%

牛牛想知道，这门课程自己最终能得到多少分。

【输入格式】

输入文件名为 score.in。

输入文件只有 1 行，包含三个非负整数 A、B、C，分别表示牛牛的作业成绩、小测成绩和期末考试成绩。相邻两个数之间用一个空格隔开，三项成绩满分都是 100 分。

【输出格式】

输出文件名为 score.out。

输出文件只有 1 行，包含一个整数，即牛牛这门课程的总成绩，满分也是 100 分。

【输入样例 1】

100 100 80

【输入样例 1】

90

【输入输出样例 1 说明】

牛牛的作业成绩是 100 分，小测成绩是 100 分，期末考试成绩是 80 分，总成绩是 $100 \times 20\% + 100 \times 30\% + 80 \times 50\% = 20 + 30 + 40 = 90$ 。

【数据说明】

对于 30% 的数据， $A = B = 0$ 。

对于另外 30% 的数据， $A = B = 100$ 。

对于 100% 的数据， $0 \leq A, B, C \leq 100$ 且 A, B, C 都是 10 的整数倍。

2.3 NOIP2008 ISBN 号码 (isbn.cpp)

【问题描述】

每一本正式出版的图书都有一个 ISBN 号码与之对应，ISBN 码包括 9 位数字、1 位识别码和 3 位分隔符，其规定格式如“x-xxx-xxxxx-x”，其中符号“-”是分隔符（键盘上的减号），最后一位是识别码，例如 0-670-82162-4 就是一个标准的 ISBN 码。ISBN 码的首位数字表示书籍的出版语言，例如 0 代表英语；第一个分隔符“-”之后的三位数字代表出版社，例如 670 代表维京出版社；第二个分隔符之后的五位数字代表该书在出版社的编号；最后一位为识别码。识别码的计算方法如下：

首位数字乘以 1 加上次位数字乘以 2……以此类推，用所得的结果 mod 11，所得的余数即为识别码，如果余数为 10，则识别码为大写字母 X。例如 ISBN 号码 0-670-82162-4 中的识别码 4 是这样得到的：对 067082162 这 9 个数字，从左至右，分别乘以 1, 2, ..., 9，再求和，即 $0 \times 1 + 6 \times 2 + \dots + 2 \times 9 = 158$ ，然后取 $158 \bmod 11$ 的结果 4 作为识别码。

你的任务是编写程序判断输入的 ISBN 号码中识别码是否正确，如果正确，则仅输出“Right”；如果错误，则输出你认为是正确的 ISBN 号码。

【输入格式】

输入只有一行，是一个字符序列，表示一本书的 ISBN 号码（保证输入符合 ISBN 号码的格式要求）。

【输出格式】

输出一行，假如输入的 ISBN 号码的识别码正确，那么输出“Right”，否则，按照规定的格式，输出正确的 ISBN 号码（包括分隔符“-”）。

【样例输入 1】

0-670-82162-4

【样例输出 1】

Right

【样例输入 2】

0-670-82162-0

【样例输出 2】

0-670-82162-4

3 专题练习—字符&字符串（每题 20-40 分钟）

3.1 字符串分解 (string.cpp)

一行文本以逗号分开，写一个函数将各段文本打印出来。

输入：hello,world,good,morning

输出：

hello
world
good
morning

3.2 破译密码 (password.cpp)。

作为著名情报员的你，截获了敌人一份加密的资料。经过研究，你发现了加密规律：

- 1) 原文中所有的字符都在字母表中被循环左移了三个位置 (dec->abz);
- 2) 逆序存储 (abcd->dcba);
- 3) 大小写反转 (abXY->ABxy)。

输入：一个加密的字符串

输出：解密后的字符串

输入样例：GS00WFAS0q

输出样例：Trvdizrrvj

3.3 乒乓球 (pingpong.cpp)

【题目描述】

华华通过以下方式进行分析，首先将比赛每个球的胜负列成一张表，然后分别计算在 11 分制和 21 分制下，双方的比赛结果（截至记录末尾）。比如现在有这份记录，（其中 W 表示华华获得一分，L 表示华华对手获得一分）：

WWWWWWWWWWWWWWWWWWWWLW

在 11 分制下，此时比赛的结果是华华第一局 11 比 0 获胜，第二局 11 比 0 获胜，正在进行第三局，当前比分 1 比 1。而在 21 分制下，此时比赛结果是华华第一局 21 比 0 获胜，正在进行第二局，比分 2 比 1。如果一局比赛刚开始，则此时比分为 0 比 0。直到分差大于或者等于 2，才一局结束。

你的程序就是要对于一系列比赛信息的输入（WL 形式），输出正确的结果。

【输入输出格式】

输入格式：

每个输入文件包含若干行字符串，字符串有大写的 W、L 和 E 组成。其中 E 表示比赛信息结束，程序应该忽略 E 之后的所有内容。

输出格式：

输出由两部分组成，每部分有若干行，每一行对应一局比赛的比分（按比赛信息输入顺序）。其中第一部分是 11 分制下的结果，第二部分是 21 分制下的结果，两部分之间由一个空行分隔。

【输入输出样例】

输入样例#1：

WWWWWWWWWWWWWWWWWW

WWLWE

输出样例#1：

11:0

11:0

1:1

21:0

2:1

【说明】

每行至多 25 个字母，最多有 2500 行

4 T2 真题（每题 40-60 分钟）

4.1 NOIP2018 龙虎斗 (fight.cpp)

【问题描述】

轩轩和凯凯正在玩一款叫《龙虎斗》的游戏，游戏的棋盘是一条线段，线段上有 n 个兵营（自左至右编号 $1 \sim n$ ），相邻编号的兵营之间相隔 1 厘米，即棋盘为长度为 $n-1$ 厘米的线段。 i 号兵营里有 c_i 位工兵。下面图 1 为 $n=6$ 的示例：

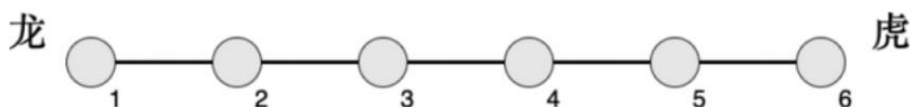


图 1. $n = 6$ 的示例

轩轩在左侧，代表“龙”；凯凯在右侧，代表“虎”。他们以 m 号兵营作为分界，靠左的工兵属于龙势力，靠右的工兵属于虎势力，而第 m 号兵营中的工兵很纠结，他们不属于任何一方。

一个兵营的气势为：该兵营中的工兵数 \times 该兵营到 m 号兵营的距离；参与游戏一方的势力定义为：属于这一方所有兵营的气势之和。

下面图 2 为 $n=6, m=4$ 的示例，其中红色为龙方，黄色为虎方：

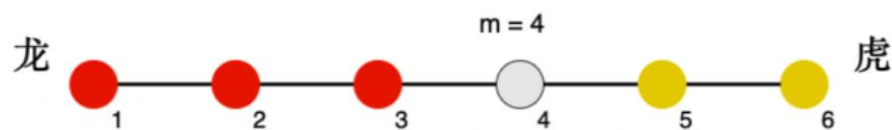


图 2. $n = 6, m = 4$ 的示例

游戏过程中，某一刻天降神兵，共有 s_1 位工兵突然出现在了 p_1 号兵营。作为轩轩和凯凯的朋友，你知道如果龙虎双方气势差距太悬殊，轩轩和凯凯就不愿意继续玩下去了。为了让游戏继续，你需要选择一个兵营 p_2 ，并将你手里的 s_2 位工兵全部派往兵营 p_2 ，使得双方气势差距尽可能小。

注意：你手中的工兵落在哪个兵营，就和该兵营中其他工兵有相同的势力归属（如果落在 m 号兵营，则不属于任何势力）

【输入输出格式】

输入格式：

输入文件的第一行包含一个正整数 n ，代表兵营的数量。

接下来的一行包含 n 个正整数，相邻两数之间以一个空格分隔，第 i 个正整数代表编号为 i 的兵营中起始时的工兵数量 c_i 。接下来的一行包含四个正整数，相邻两数间以一个空格分隔，分别代表 m, p_1, s_1, s_2

输出格式：

输出文件有一行，包含一个正整数，即 p_2 ，表示你选择的兵营编号。如果存在多个编号同时满足最优，取最小的编号。

【输入输出样例 1】

fight. in	fight. out
6	2
2 3 2 3 2 3	
4 6 5 2	

【输入输出样例 2】

fight. in	fight. out
6	1
1 1 1 1 1 16	
5 4 1 1	

【输入输出样例 1 说明】

见问题描述中的图 2。双方以 $m=4$ 号兵营分界，有 $s_1=5$ 位工兵突然出现在 $p_1 = 6$ 号兵营。

龙方的气势为：

$$2 \times (4-1) + 3 \times (4-2) + 2 \times (4-3) = 14$$

虎方的气势为：

$$2 \times (5-4) + (3+5) \times (6-4) = 18$$

当你将手中的 $s_2 = 2$ 位工兵派往 $p_2 = 2$ 号兵营时，龙方的气势变为：

$$14 + 2 \times (4-2) = 18$$

此时双方气势相等。

【输入输出样例 2 说明】

双方以 $m = 5$ 号兵营分界，有 $s_1 = 1$ 位工兵突然出现在 $p_1 = 4$ 号兵营。

龙方的气势为：

$$1 \times (5-1) + 1 \times (5-2) + 1 \times (5-3) + (1+1) \times (5-4) = 11$$

虎方的气势为：

$$16 \times (6-5) = 16$$

当你将手中的 $s_2 = 1$ 位工兵派往 $p_2 = 1$ 号兵营时，龙方的气势变为：

$$11 + 1 \times (5-1) = 15$$

此时可以使双方气势的差距最小。

【数据规模与约定】

$$1 < m < n, 1 \leq p_1 \leq n$$

对于 20% 的数据, $n = 3, m = 2, c_i = 1, s_1, s_2 \leq 100$

另 20% 的数据, $n \leq 10, p_1 = m, c_i = 1, s_1, s_2 \leq 100$

对于 60% 的数据, $n \leq 100, c_i = 1, s_1, s_2 \leq 100 \leq 100$

对于 80% 的数据, $n \leq 100, c_i, s_1, s_2 \leq 100 \leq 100$

对于 100% 的数据, $n \leq 10^5, c_i, s_1, s_2 \leq 10^9$

5 算法挑战—贪心（尝试）

5.1 互不相交区间问题

活动安排 (plan.cpp)

【问题描述】

设有 n 个活动的集合 $E = \{1, 2, \dots, n\}$, 其中每个活动都要求使用同一资源, 如演讲会场等, 而在同一时间内只有一个活动能使用这一资源。每个活动 i 都有一个要求使用该资源的起始时间 s_i 和一个结束时间 f_i , 且 $s_i < f_i$ 。如果选择了活动 i , 则它在半开区间 $[s_i, f_i)$ 内占用资源。若区间 $[s_i, f_i)$ 与区间 $[s_j, f_j)$ 不相交, 则称活动 i 与活动 j 是相容的。也就是说, 当 $s_i \geq f_j$ 或 $s_j \geq f_i$ 时, 活动 i 与活动 j 相容。选择出由互相兼容的活动组成的最大集合。

【输入格式】

第一行一个整数 n ($n \leq 1000$) ; 接下来的 n 行, 每行两个整数 s_i 和 f_i 。

【输出格式】

输出尽可能多的互相兼容的活动个数

【输入样例】

```
4
1 3
4 6
2 5
1 7
```

【输出样例】

```
2
```

5.2 区间选点问题

种树 (tree.cpp)

【问题描述】

一条街道的一边有几座房子, 因为环保原因居民想要在路边种些树, 路边的地区被分割成 n 块, 并被编号为 $1 \dots n$, 每块大小为一个单位尺寸并最多可以种一棵树, 每个居民想在门前种些树并指定了三个数 b, e, t 这三个数分别表示该居民想在 b 和 e 之间最少种 t 棵树, 当然, $b \leq e, t \leq e - b + 1$, 允许居民想种树的子区域可以交叉。出于资金紧缺的原因, 环保部门请你求出能满足所有居民的种树要求时所需树的最少数量。

【输入格式】

第一行为 n , 表示区域的个数

第二行为 h , 表示房子的数目

下面 h 行描述居民的需要: b, e, t ($0 < b \leq e \leq 30000, t \leq e - b + 1$) 分别用空格分开

【输出格式】

输出一个数, 为满足所有居民的要求, 所需要种树的最少数量。

【输入样例】

```
9
4
1 4 2
4 6 2
8 9 2
3 5 2
```

【输出样例】

```
5
```

6 T3 真题（尝试）

N01P2017 棋盘 (chess.cpp)

【问题描述】

有一个 $m \times m$ 的棋盘，棋盘上每一个格子可能是红色、黄色或没有任何颜色的。你现在要从棋盘的最左上角走到棋盘的最右下角。

任何一个时刻，你所站在的位置必须是有颜色的（不能是无色的），你只能向上、下、左、右四个方向前进。当你从一个格子走向另一个格子时，如果两个格子的颜色相同，那你不需要花费金币；如果不同，则你需要花费 1 个金币。

另外，你可以花费 2 个金币施展魔法让下一个无色格子暂时变为你指定的颜色。但这个魔法不能连续使用，而且这个魔法的持续时间很短，也就是说，如果你使用了这个魔法，走到了这个暂时有颜色的格子上，你就不能继续使用魔法；只有当你离开这个位置，走到一个本来就有颜色的格子上的时候，你才能继续使用这个魔法，而当你离开了这个位置（施展魔法使得变为有颜色的格子）时，这个格子恢复为无色。现在你要从棋盘的最左上角，走到棋盘的最右下角，求花费的最少金币是多少？

【输入格式】

输入文件名为 chess.in。

数据的第一行包含两个正整数 m, n ，以一个空格分开，分别代表棋盘的大小，棋盘上有颜色的格子的数量。接下来的 n 行，每行三个正整数 x, y, c ，分别表示坐标为 (x, y) 的格子有颜色 c 。其中 $c=1$ 代表黄色， $c=0$ 代表红色。相邻两个数之间用一个空格隔开。棋盘左上角的坐标为 $(1, 1)$ ，右下角的坐标为 (m, m) 。棋盘上其余的格子都是无色。保证棋盘的左上角，也就是 $(1, 1)$ 一定是有颜色的。

【输出格式】

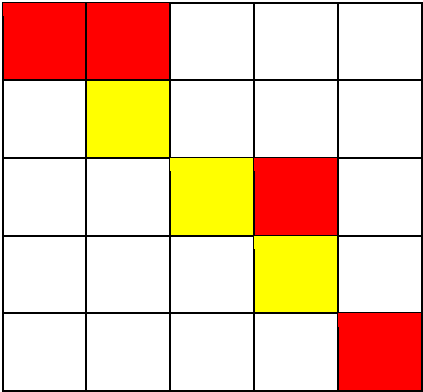
输出文件名为 chess.out。

输出一行，一个整数，表示花费的金币的最小值，如果无法到达，输出-1。

【输入输出样例 1】

chess.in	chess.out
5 7 1 1 0 2 0 2 1 3 1 4 0 4 1 5 0	8

【输入输出样例 1 说明】



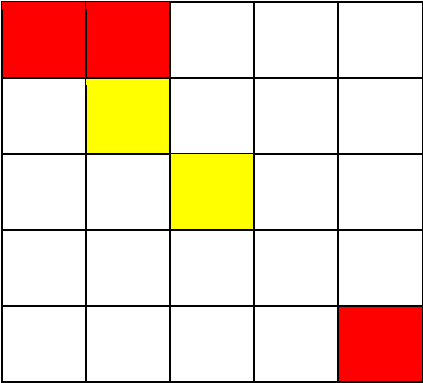
从 $(1, 1)$ 开始，走到 $(1, 2)$ 不花费金币从 $(1, 2)$ 向下走到 $(2, 2)$ 花费 1 枚金币从 $(2, 2)$ 施展魔法，将 $(2, 3)$ 变为黄色，花费 2 枚金币从 $(2, 2)$ 走到 $(2, 3)$ 不花费金币从 $(2, 3)$ 走到 $(3, 3)$ 不花费金币从 $(3, 3)$ 走到 $(3, 4)$ 花费 1 枚金币从 $(3, 4)$ 走到 $(4, 4)$ 花费 1 枚金币从 $(4, 4)$ 施展魔法，将 $(4, 5)$ 变为黄色，花费 2 枚金币，从 $(4, 4)$ 走到 $(4, 5)$ 不花费金币从 $(4, 5)$ 走到 $(5, 5)$ 花费 1 枚金币共花费 8 枚金币。

【输入输出样例 2】

chess.in	chess.out
----------	-----------

5 5 1 1 0 2 0 2 1 3 1 5 5 0	-1
--	----

【输入输出样例 2 说明】



从 (1, 1) 走到 (1, 2)，不花费金币从 (1, 2) 走到 (2, 2)，花费 1 金币施展魔法将 (2, 3) 变为黄色，并从 (2, 2) 走到 (2, 3) 花费 2 金币从 (2, 3) 走到 (3, 3) 不花费金币 从 (3, 3) 只能施展魔法到达 (3, 2)，(2, 3)，(3, 4)，(4, 3)，而从以上四点均无法到达 (5, 5)，故无法到达终点，输出 -1。

【数据规模与约定】

- 对于 30%的数据， $1 \leq m \leq 5$ ， $1 \leq n \leq 10$ 。
- 对于 60%的数据， $1 \leq m \leq 20$ ， $1 \leq n \leq 200$ 。
- 对于 100%的数据， $1 \leq m \leq 100$ ， $1 \leq n \leq 1,000$ 。