

乐享未来·青少年编程

2019NOIP 精英营习题集

精英营习题集	1
1 模板程序	3
1.1 判断素数 (prime.cpp)	3
1.2 最大公约数最小公倍数 (gcd.cpp)	3
1.3 数字翻转 (reverse.cpp)	3
1.4 字符串翻转 (strReverse.cpp)	3
1.5 判断闰年 (leap.cpp)	3
1.6 大小写转换 (toUpper.cpp)	3
1.7 一个数的 n 次幂 (pow.cpp)	3
1.8 斐波那契数列 (fibo.cpp)	3
2 基础练习题	3
2.1 求表达式值 (calc.cpp)	3
2.2 温度转换 (temperature.cpp)	3
2.3 大象喝水 (drink.cpp)	3
2.4 三位数各位之和 (three.cpp)	4
2.5 两数之差 (sub.cpp)	4
2.6 买水果 (fruit.cpp)	4
2.7 奥运金牌计数 (gold.cpp)	4
2.8 三角形判断 (triangle.cpp)	5
2.9 收集瓶盖赢大奖 (prize.cpp)	5
2.10 计算邮资 (fee.cpp)	5
2.11 最高分数 (top.cpp)	5
2.12 生日 (birthday.cpp)	6
2.13 余数相同问题 (same.cpp)	6
2.14 与 7 无关的数 (7.cpp)	6
2.15 字符串分解 (string.cpp)	7
2.16 年龄排序 (ageSort.cpp)	7
2.17 身高问题 (hight.cpp)	7
3 进阶练习题	8
3.1 作业统计 (homework.cpp)	8
3.2 数字三角形 (numTriangle.cpp)	8
3.3 回型方阵 (hxfz.cpp)	8
3.4 破译密码 (password.cpp)	8
3.5 校门外的树 (tree.cpp)	9
3.6 乒乓球 (pingpong.cpp)	9
3.7 津津的储蓄计划 (save.cpp)	9
4 T1 真题	10
4.1 NOIP2018 标题统计 (title.cpp)	10
4.2 NOIP2017 成绩 (score.cpp)	11
4.3 NOIP2016 买铅笔 (pen.cpp)	11
4.4 NOIP2015 金币 (coin.cpp)	11
4.5 NOIP2014 珠心算测验 (count.cpp)	12
4.6 NOIP2013 计数问题 (count.cpp)	12
4.7 NOIP2012 质因数分解 (prime.cpp)	12
4.8 NOIP2011 数字反转 (reverse.cpp)	13
4.9 NOIP2010 数字统计 (two.cpp)	13
4.10 NOIP2009 多项式输出 (poly.cpp)	13
4.11 NOIP2008 ISBN 号码 (isbn.cpp)	14
4.12 NOIP2007 奖学金 (scholar.cpp)	14
5 T2 真题	15
5.1 NOIP2018 龙虎斗 (fight.cpp)	15

5.2	NOIP2017 图书管理员 (librarian.cpp)	17
5.3	NOIP2016 日期回文 (date.cpp)	17
5.4	NOIP2015 扫雷游戏 (mine.cpp)	18
5.5	NOIP2014 比例简化 (ratio.cpp)	18
5.6	NOIP2013 表达式求值 (expr.cpp)	19
5.7	NOIP2012 寻宝 (treasure.cpp)	19
5.8	NOIP2011 统计单词数 (stat.cpp)	20
5.9	NOIP2010 接水问题 (water.cpp)	20
5.10	NOIP2009 分数线划定 (sorce.cpp)	21
5.11	NOIP2008 排座椅 (seat.cpp)	21
6	算法挑战	22
6.1	进制转换 (change.cpp)	22
6.2	数列 (series.cpp)	22
6.3	高精度乘法 (mult.cpp)	22
6.4	蚱蜢 (grasshopper.cpp)	22
6.5	遭遇战 (battle.cpp)	23
6.6	乒乓球 (pingpong.cpp)	23
6.7	铺瓷砖 (tile)	24
6.8	城市路径 (route.cpp)	24
6.9	新约瑟夫问题 (newJoseph.cpp)	24
6.10	找伪币 (coin.cpp)	25
6.11	循环比赛日程表 (schedule.cpp)	25
6.12	取模运算 (model.cpp)	25
6.13	地毯填铺 (carpet.cpp)	25
6.14	记忆化斐波那契数列 (fibo.cpp)	26
6.15	砍伐树木 (tree.cpp)	26
6.16	独木舟 (canoe.cpp)	27
6.17	最大整数 (maxInt.cpp)	27
6.18	排队接水 (water.cpp)	27
6.19	楼层编号 (floor.cpp)	28
6.20	最大黑色区域 (area.cpp)	28
6.21	关系网络 (relationship.cpp)	28
6.22	溶液模拟器 (simulator.cpp)	29
6.23	背包问题 (pack.cpp)	29
6.24	数的拆分 (number.cpp)	30
6.25	导弹拦截 (missile.cpp)	30
7	T3 真题	30
7.1	NOIP2018 摆渡车 (bus.cpp)	30
7.2	NOIP2017 棋盘 (chess.cpp)	31
7.3	NOIP2016 海港 (port.cpp)	33
7.4	NOIP2015 求和 (sum.cpp)	34
7.5	NOIP2014 螺旋矩阵 (spiral.cpp)	34

1 模板程序

- 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 判断闰年 (leap.cpp)
- 1.6 大小写转换 (toUpper.cpp)
- 1.7 一个数的 n 次幂 (pow.cpp)
- 1.8 斐波那契数列 (fibo.cpp)

2 基础练习题

2.1 求表达式值 (calc.cpp)

【题目描述】

输入 a 和 b 两个整数的值，输出表达式 $3a-2b+1$ 的值。

【输入输出格式】

输入格式：

输入两个正整数 a 和 b

输出格式：

输出一个正整数，表示表达式的值

【输入输出样例】

输入样例#1：

3 1

输出样例#1：

8

2.2 温度转换 (temperature.cpp)

【题目描述】

利用公式 $C=5*(F-32)/9$ (其中 C 表示摄氏温度, F 表示华氏温度) 进行计算转化, 输入华氏温度 F, 输出摄氏温度 C。

【输入输出格式】

输入格式：

输入一个正整数 F

输出格式：

输出一个正整数 C

【输入输出样例】

输入样例#1：

41

输出样例#1：

5

2.3 大象喝水 (drink.cpp)

【题目描述】

一只大象口渴了, 要喝 n 桶水才能解渴, 但现在只有一个深 h 厘米, 底面半径为 r 厘米的小圆桶 (h 和 r 都是整数), 问大象一天要喝多少升水才会解渴。

【输入输出格式】

输入格式：

输入一行数据, 包括三个整数, 以一个空格分开, 分别表示大象需要喝水的桶数, 小圆桶的深 h 和底面半径 r, h 和 r 的单位都是厘米。

输出格式：

输出一个整数, 表示大象一天要喝多少升水才会解渴。

【输入输出样例】

输入样例#1：

3 16 7

输出样例#1：

7.38528

2.4 三位数各位之和 (three.cpp)

【题目描述】

给出任意一个三位数，求这个三位数的三个数字之和。例如输入 897，输出 24 ($8+9+7=24$)。

【输入输出格式】

输入格式：

1 行，输入一个三位数。

输出格式：

输出一个数字，这个数字表示三位数的三个数字之和。

【输入输出样例】

输入样例#1：

897

输出样例#1：

24

2.5 两数之差 (sub.cpp)

【题目描述】

给出任意一个四位数，将这个数的各位数字倒序合成一个新数，输出原数和新数的差。

【输入输出格式】

输入格式：

1 行，输入一个四位数。

输出格式：

输出一个数字，这个数字表示原数和新数的差。

【输入输出样例】

输入样例#1：

1234

输出样例#1：

3087

2.6 买水果 (fruit.cpp)

【题目描述】

超市里有香蕉、苹果、葡萄三种水果，它们分别的单价为 a 、 b 、 c 元，如果只有 n 元，并且只能购买其中一种水果整数斤，请计算并输出三种水果分别可以买几斤，剩余多少钱。

【输入输出格式】

输入格式：

输入两行数据，第一行包括四整数，以一个空格分开，分别表示香蕉、苹果、葡萄的单价，第二行包括一个整数，表示购买水果的金钱数。

输出格式：

输出三行数据，分别包括两个整数，第一行表示可以购买香蕉的斤数和剩余的金钱数；第二行表示可以购买苹果的斤数和剩余的金钱数；第三行表示可以购买葡萄的斤数和剩余的金钱数。

【输入输出样例】

输入样例#1：

5 8 12

39

输出样例#1：

7 4

4 7

3 3

2.7 奥运金牌计数 (gold.cpp)

【题目描述】

2008 年北京奥运会，A 国的运动员参与了 n 天的决赛项目 ($1 \leq n \leq 17$)。现在要统计一下 A 国所获得的金、银、铜牌数目及总奖牌数。输入第 1 行是 A 国参与决赛项目的天数 n ，其后 n 行，每一行是该国某一天获得的金、银、铜牌数目。输出是 4 个整数，为 A 国所获得的金、银、铜牌数目及总奖牌数。

【输入输出格式】

输入格式：

有 $n+1$ 行，第 1 行是 A 国参与决赛项目的天数 n ，其后 n 行，每一行是该国某一天获得的金、银、铜牌数目，数目之间以一个空格分开。

输出格式：

输出一行数据，包括 4 个整数，为 A 国所获得的金、银、铜牌数目及总奖牌数，数目之间以一个空格分开。

【输入输出样例】

输入样例#1:

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

输出样例#1:

```
4 4 3 11
```

2.8 三角形判断 (triangle.cpp)

【题目描述】

给定三个正整数，分别表示三条线段的长度，判断这三条线段能否构成一个三角形，如果能构成三角形，则输出“**Yes**”，否则输出“**No**”。

【输入输出格式】

输入格式:

1 行，包含三个正整数，分别表示三条线段的长度，数与数之间以一个空格分开。

输出格式:

输出一行数据，如果能构成三角形，则输出“**Yes**”，否则输出“**No**”

【输入输出样例】

输入样例#1:

```
3 4 5
```

输出样例#1:

```
Yes
```

2.9 收集瓶盖赢大奖 (prize.cpp)

【题目描述】

某饮料公司最近推出了一个“收集瓶盖赢大奖”的活动：如果你拥有的 10 个印有“幸运”或 20 个印有“鼓励”的瓶盖，就可以兑换一个神秘大奖。现分别给出你拥有的印有“幸运”和“鼓励”的瓶盖数，判断是否可以去兑换大奖。若可以兑换大奖，输出 1；否则输出 0。

【输入输出格式】

输入格式:

输入一行，包含两个整数，分别是印有“幸运”和“鼓励”的瓶盖数，用一个空格隔开。

输出格式:

输出一个正整数，若可以兑换大奖，输出 1；否则输出 0。

【输入输出样例】

输入样例#1:

```
11 19
```

输出样例#1:

```
1
```

2.10 计算邮资 (fee.cpp)

【题目描述】

根据邮件的质量和用户是否选择加急计算邮费。计算规则：质量在 1000 克以内的（包括 1000 克），基本费 8 元；超过 1000 克的部分，每 500 克加收超重费 4 元，不足 500 克部分按 500 克计算；如果用户选择加急，多收 5 元。

【输入输出格式】

输入格式:

输入一行，包含一个整数和一个字符，以空格分开，分别表示质量（单位为克）和是否加急，如果是字符 y，说明选择加急，如果是字符 n，说明不选择加急。

输出格式:

输出一行，包含一个整数，表示邮费。

【输入输出样例】

输入样例#1:

```
1200 y
```

输出样例#1:

```
17
```

2.11 最高分数 (top.cpp)

【题目描述】

孙老师讲授《计算概论》这门课的期中考试刚刚结束，他想知道考试中取得的最高分数是多少分，由于人数比较多，

他觉得这件事情交给计算机来做比较方便。你能帮助孙老师解决这个问题吗？

【输入输出格式】

输入格式：

有 2 行，第 1 行是有一个整数 n ($1 \leq n \leq 100$)，表示参加这次考试的人数，第 2 行，是这 n 个学生的成绩，相邻两个数之间用单个空格隔开，所有成绩均为 $0 \sim 100$ 之间的整数。

输出格式：

输出一行数据，包括 1 个整数，为所求的最高分数。

【输入输出样例】

输入样例#1：

5

85 73 92 78 89

输出样例#1：

92

2.12 生日 (birthday.cpp)

【题目描述】

cjf 君想调查学校 01 组每个同学的生日，并按照从大到小的顺序排序。但 cjf 君最近作业很多，没有时间，所以请你帮她排序。

【输入输出格式】

输入格式：

有 2 行，

第 1 行为 01 组总人数 n ；

第 2 行至第 $n+1$ 行分别是每人的姓名 s 、出生年 y 、月 m 、日 d 。

输出格式：

有 n 行，

即 n 个生日从大到小同学的姓名。(如果有两个同学生日相同，输入靠后的同学先输出)

【输入输出样例】

输入样例#1：

3

Yangchu 1992 4 23

Qiujiangya 1993 10 13

Luowen 1991 8 1

输出样例#1：

Luowen

Yangchu

Qiujiangya

【数据说明】

数据规模 $1 < n < 100$ $\text{length}(s) < 20$

2.13 余数相同问题 (same.cpp)

【题目描述】

已知三个正整数 a, b, c ，现有一个大于 1 的整数 x ，将其作为除数分别除 a, b, c ，得到的余数相同。请问满足上述条件的 x 的最小值是多少？数据保证 x 有解。

【输入输出格式】

输入格式：

有 1 行，三个不大于 1000000 的正整数 a, b, c ，每个整数之间用一个空格隔开。

输出格式：

输出一行数据，包括 1 个整数，即满足条件的 x 的最小值。

【输入输出样例】

输入样例#1：

300 262 205

输出样例#1：

19

2.14 与 7 无关的数 (7.cpp)

【问题描述】

一个正整数，如果它能被 7 整除，或者它的某一位上的数字为 7，则称其为“与 7 相关”的数。

现在，请编程求出所有小于或等于 n 的“与 7 无关”的正整数个数。

【输入格式】

一行一个正整数 n , $n \leq 10^6$ 。

【输出格式】

一行一个整数，表示答案。

【样例输入】

21

【样例输出】

17

2.15 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
```

2.16 年龄排序 (ageSort.cpp)

【问题描述】

输入 n 个学生的信息，包括姓名、性别、出生年月。要求按年龄从小到大依次输出这些学生的信息。数据保证没有学生同年同月出生。

【输入格式】

第一行一个整数 n ，表示学生人数， $n \leq 100$ 。

接下来 n 行，每一行依次输入学生的姓名、性别、出生年份、出生月份。

【输出格式】

按年龄从小到大，一行输出一个学生的原始信息。

【输入样例】

```
5
John male 1999 12
David female 1999 8
Jason male 1998 11
Jack female 1998 8
Kitty female 2000 7
```

【输出样例】

```
Kitty female 2000 7
John male 1999 12
David female 1999 8
Jason male 1998 11
Jack female 1998 8
```

2.17 身高问题 (hight.cpp)

【问题描述】

输入 n 个学生的信息，每个学生信息包括姓名、身高、学号。编程输出身高最高的学生的信息。

【输入格式】

第 1 行一个正整数 n ，表示学生个数， $n \leq 100$ 。

以下 n 行，每一行依次输入学生的姓名、身高、学号。

【输出格式】

输出最高的学生信息，如存在身高一样的请输出学号小的那个同学。

【输入样例】

```
5
John 172 20160302
David 173 20160306
Jason 168 20160309
Jack 152 20160311
Kitty 147 20160319
```

【输出样例】

```
David 173 20160306
```

3 进阶练习题

3.1 作业统计 (homework.cpp)

【问题描述】

为了解学生的课后作业负担情况，需要统计学生连续若干天完成作业所需的总时间。现在，输入某位学生 n 天完成作业的时间，格式为时、分、秒，最后输出这位学生 n 天完成作业的总时间（秒）。

【输入格式】

第 1 行一个正整数 n ，表示有 n 天；

第 2~第 $n+1$ 行，每行 3 个整数，分别代表时、分、秒。

【输出格式】

一行信息，表示这个学生完成作业的总时间，具体格式见输出样例。

【输入样例】

```
3
1 20 30
1 20 45
1 19 30
```

【输出样例】

```
4hour 0minute 45second
```

3.2 数字三角形 (numTriangle.cpp)

【问题描述】

输入一个正整数 n ，输出 n 行的数字三角形。其中，第 1 行为数字 1，第 2 行为数字 23，第 3 行为数字 456，第 4 行为数字 7890，第 5 行为数字 12345，...

【输入格式】

一行一个正整数 n ， $1 \leq n \leq 100$ 。

【输出格式】

n 行的数字三角形。

【样例输入】

```
4
```

【样例输出】

```
1
23
456
7890
```

3.3 回型方阵 (hxfz.cpp)

【问题描述】

输入一个正整数 n ，输出 $n \times n$ 的回型方阵。例如， $n=5$ 时，输出：

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

【输入格式】

一行一个正整数 n ， $2 \leq n \leq 9$ 。

【输出格式】

共 n 行，每行包含 n 个正整数，之间用一个空格隔开。

3.4 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

3.5 校门外的树 (tree.cpp)

【题目描述】

校门外的马路上本来从编号 0 到 L，每一编号的位置都有 1 棵树。有砍树者每次从编号 A 到 B 处连续砍掉每 1 棵树，就连树苗也不放过（记 0 A B，含 A 和 B）；幸运的是还有植树者每次从编号 C 到 D 中凡是空穴（树被砍且还没种上树苗或树苗又被砍掉）的地方都补种上树苗（记 1 C D，含 C 和 D）；问最终校门外的树苗多少棵？植树者种上又被砍掉的树苗有多少棵？

【输入输出格式】

输入格式：

第一行 L 和 N，表示校园外原来有 L+1 棵树，并有 N 次砍树或种树的操作。

以下 N 行，砍树或植树的标记和范围，每行 3 个整数。

L (1 ≤ L ≤ 10000) 和 N (1 ≤ N ≤ 100)

输出格式：

共两行。第 1 行校门外的树苗数目，第 2 行种上又被拔掉的树苗数目。

【输入输出样例】

输入样例#1：

```
10 3
0 2 6
1 1 8
0 5 7
```

输出样例#1：

```
3
2
```

3.6 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）：

WWWWWWWWWWWWWWWWWWLW

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：

```
WWWWWWWWWWWWWWWWWWLW
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

3.7 津津的储蓄计划 (save.cpp)

【问题描述】

津津的零花钱一直都是自己管理。每个月的月初妈妈给津津 300 元钱，津津会预算这个月的花销，并且总能做到实际花销和预算的相同。

为了让津津学习如何储蓄，妈妈提出，津津可以随时把整百的钱存在她那里，到了年末她会加上 20% 还给津津。因此津津制定了一个储蓄计划：每个月的月初，在得到妈妈给的零花钱后，如果她预计到这个月的月末手中还会有多于

100 元或恰好 100 元，她就会把整百的钱存在妈妈那里，剩余的钱留在自己手中。

例如 11 月初津津手中还有 83 元，妈妈给了津津 300 元。津津预计 11 月的花销是 180 元，那么她就会在妈妈那里存 200 元，自己留下 183 元。到了 11 月月末，津津手中会剩下 3 元钱。

津津发现这个储蓄计划的主要风险是，存在妈妈那里的钱在年末之前不能取出。有可能在某个月的月初，津津手中的钱加上这个月妈妈给的钱，不够这个月的原定预算。如果出现这种情况，津津将不得不在这个月省吃俭用，压缩预算。

现在请你根据 2004 年 1 月到 12 月每个月津津的预算，判断会不会出现这种情况。如果不会，计算到 2004 年年末，妈妈将津津平常存的钱加上 20% 还给津津之后，津津手中会有多少钱。

【输入文件】

输入文件 `save.in` 包括 12 行数据，每行包含一个小于 350 的非负整数，分别表示 1 月到 12 月津津的预算。

【输出文件】

输出文件 `save.out` 包括一行，这一行只包含一个整数。如果储蓄计划实施过程中出现某个月钱不够用的情况，输出 `-X`，`X` 表示出现这种情况的第一个月；否则输出到 2004 年年末津津手中会有多少钱。

【样例输入 1】

290
230
280
200
300
170
340
50
90
80
200
60

【样例输出 1】

-7

【样例输入 2】

290
230
280
200
300
170
330
50
90
80
200
60

【样例输出 2】

1580

4 T1 真题

4.1 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。

4.2 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。

4.3 NOIP2016 买铅笔(pen.cpp)

【问题描述】

P 老师需要去商店买 n 支铅笔作为小朋友们参加 NOIP 的礼物。她发现商店一共有 3 种包装的铅笔，不同包装内的铅笔数量有可能不同，价格也有可能不同。为了公平起见，P 老师决定只买同一种包装的铅笔。

商店不允许将铅笔的包装拆开，因此 P 老师可能需要购买超过 n 支铅笔才够给小朋友们发礼物。

现在 P 老师想知道，在商店每种包装的数量都足够的情况下，要买够至少 n 支铅笔最少需要花费多少钱。

【输入描述】

输入的第一行包含一个正整数 n ，表示需要的铅笔数量。

接下来三行，每行用两个正整数描述一种包装的铅笔：其中第一个整数表示这种包装内铅笔的数量，第二个整数表示这种包装的价格。

保证所有的 7 个数都是不超过 10000 的正整数

【输出描述】

输出一行一个整数，表示 P 老师最少需要花费的钱。

【样例输入】

57
2 2
50 30
30 27

【样例输出】

54

4.4 NOIP2015 金币(coin.cpp)

【题目描述】

国王将金币作为工资，发放给忠诚的骑士。第一天，骑士收到一枚金币；之后两天（第二天和第三天），每天收到两枚金币；之后三天（第四、五、六天），每天收到三枚金币；之后四天（第七、八、九、十天），每天收到四枚金币……；这种工资发放模式会一直这样延续下去：当连续 N 天每天收到 N 枚金币后，骑士会在之后的连续 $N+1$ 天里，每天收到 $N+1$ 枚金币。

请计算在前 K 天里，骑士一共获得了多少金币。

输入格式：输入文件只有 1 行，包含一个正整数 K ，表示发放金币的天数。

输出格式：输出文件只有 1 行，包含一个正整数，即骑士收到的金币数。

【输入样例】

6

【输出样例】

14

【输入样例】

1000

【输出样例】

29820

【输入输出样例 1 说明】

骑士第一天收到一枚金币；第二天和第三天，每天收到两枚金币；第四、五、六天，每天收到三枚金币。因此一共收到 $1+2+2+3+3+3=14$ 枚金币。对于 100% 的数据， $1 \leq K \leq 10,000$ 。

4.5 NOIP2014 珠心算测验(count.cpp)

【题目描述】

珠心算是一种通过在脑中模拟算盘变化来完成快速运算的一种计算技术。珠心算训练，既能够开发智力，又能够为日常生活带来很多便利，因而在很多学校得到普及。某学校的珠心算老师采用一种快速考察珠心算加法能力的测验方法。他随机生成一个正整数集合，集合中的数各不相同，然后要求学生回答：其中有多少个数，恰好等于集合中另外两个（不同的）数之和？最近老师出了一些测验题，请你帮忙求出答案。

【输入输出格式】

输入格式：输入共两行，第一行包含一个整数 n ，表示测试题中给出的正整数个数。第二行有 n 个正整数，每两个正整数之间用一个空格隔开，表示测试题中给出的正整数。

输出格式：输出共一行，包含一个整数，表示测验题答案。

【输入样例】

4

1 2 3 4

【输出样例】

2

【样例说明】

由 $1+2=3$ ， $1+3=4$ ，故满足测试要求的答案为 2。注意，加数和被加数必须是集合中的两个不同的数。

【数据说明】

对于 100% 的数据， $3 \leq n \leq 100$ ，测验题给出的正整数大小不超过 10,000。

4.6 NOIP2013 计数问题(count.cpp)

【题目描述】

试计算在区间 1 到 n 的所有整数中，数字 x ($0 \leq x \leq 9$) 共出现了多少次？例如，在 1 到 11 中，即在 1、2、3、4、5、6、7、8、9、10、11 中，数字 1 出现了 4 次。

【输入输出格式】

输入格式：输入文件名为 count.in。输入共 1 行，包含 2 个整数 n 、 x ，之间用一个空格隔开。

输出格式：输出文件名为 count.out。输出共 1 行，包含一个整数，表示 x 出现的次数。

【输入样例】

11 1

【输出样例】

4

【说明】

对于 100% 的数据， $1 \leq n \leq 1,000,000$ ， $0 \leq x \leq 9$ 。

4.7 NOIP2012 质因数分解(prime.cpp)

【问题描述】

质因数分解，已知正整数 n 是两个不同的质数的乘积，试求出较大的那个质数。

【输入】

输入文件名为 prime.in。输入只有一行，包含一个正整数 n 。

【输出】

输出文件名为 `prime.out`。输出只有一行，包含一个正整数 p ，即较大的那个质数。

【输入输出样例】

```
prime.in prime.out
21      7
```

【数据范围】

对于 60% 的数据， $6 \leq n \leq 1000$ 。对于 100% 的数据， $6 \leq n \leq 2 \times 10^9$ 。

4.8 NOIP2011 数字反转 (reverse.cpp)

【题目描述】

给定一个整数，请将该数各个位上数字反转得到一个新数。新数也应满足整数的常见形式，即除非给定的原数为零，否则反转后得到的新数的最高位数字不应为零。

【输入输出格式】

输入格式：输入文件名为 `reverse.in`。输入共 1 行，一个整数 N 。

输出格式：输出文件名为 `reverse.out`。输出共 1 行，一个整数，表示反转后的新数。

【样例输入 1】

123

【样例输入 2】

-380

【样例输出 1】

321

【样例输出 2】

-83

4.9 NOIP2010 数字统计 (two.cpp)

【题目描述】

请统计某个给定范围 $[L, R]$ 的所有整数中，数字 2 出现的次数。比如给定范围 $[2, 22]$ ，数字 2 在数 2 中出现了 1 次，在数 12 中出现 1 次，在数 20 中出现 1 次，在数 21 中出现 1 次，在数 22 中出现 2 次，所以数字 2 在该范围内一共出现了 6 次。

【输入输出格式】

输入格式：输入文件名为 `two.in`。输入共 1 行，为两个正整数 L 和 R ，之间用一个空格隔开。

输出格式：输出文件名为 `two.out`。输出共 1 行，表示数字 2 出现的次数。

【输入样例 1】

2 22

【输入样例 2】

2 100

【输出样例 1】

6

【输出样例 2】

20

【说明】

$1 \leq L \leq R \leq 100000$ 。

4.10 NOIP2009 多项式输出 (poly.cpp)

【题目描述】

一元 n 次多项式可用如下的表达式表示：其中， $a_i x^i$ 称为 i 次项， a_i 称为 i 次项的系数。给出一个一元多项式各项的次数和系数，请按照如下规定的格式要求输出该多项式：

1. 多项式中自变量为 x ，从左到右按照次数递减顺序给出多项式。
2. 多项式中只包含系数不为 0 的项。
3. 如果多项式 n 次项系数为正，则多项式开头不出现“+”号，如果多项式 n 次项系数为负，则多项式以“-”号开头。
4. 对于不是最高次的项，以“+”号或者“-”号连接此项与前一项，分别表示此项系数为正或者系数为负。紧跟一个正整数，表示此项系数的绝对值（如果一个高于 0 次的项，其系数的绝对值为 1，则无需输出 1）。如果 x 的指数大于 1，则接下来紧跟的指数部分的形式为“ x^b ”，其中 b 为 x 的指数；如果 x 的指数为 1，则接下来紧跟的指数部分形式为“ x ”；如果 x 的指数为 0，则仅需输出系数即可。

5. 多项式中，多项式的开头、结尾不含多余的空格。

【输入输出格式】

输入格式：输入共有 2 行，第一行 1 个整数， n ，表示一元多项式的次数。第二行有 $n+1$ 个整数，其中第 i 个整数表示第 $n-i+1$ 次项的系数，每两个整数之间用空格隔开。

输出格式：输出共 1 行，按题目所述格式输出多项式。

【输入样例】

```
5
100 -1 1 -3 0 10
```

【输出样例】

```
100x^5-x^4+x^3-3x^2+10
```

【输入样例】

```
3
-50 0 0 1
```

【输出样例】

```
-50x^3+1
```

4.11 NOI P2008 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-xxxx-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
```

4.12 NOI P2007 奖学金 (scholar.cpp)

【问题描述】

某小学最近得到了一笔赞助，打算拿出其中一部分为学习成绩优秀的前 5 名学生发奖学金。期末，每个学生都有 3 门课的成绩：语文、数学、英语。先按总分从高到低排序，如果两个同学总分相同，再按语文成绩从高到低排序，如果两个同学总分和语文成绩都相同，那么规定学号小的同学排在前面，这样，每个学生的排序是唯一确定的。

任务：先根据输入的 3 门课的成绩计算总分，然后按上述规则排序，最后按排名顺序输出前 5 名学生的学号和总分。注意，在前 5 名同学中，每个人的奖学金都不相同，因此，你必须严格按上述规则排序。例如，在某个正确答案中，如果前两行的输出数据（每行输出两个数：学号、总分）是：

```
7 279
```

```
5 279
```

这两行数据的含义是：总分最高的两个同学的学号依次是 7 号、5 号。这两名同学的总分都是 279（总分等于输入的语文、数学、英语三科成绩之和），但学号为 7 的学生语文成绩更高一些。如果你的前两名的输出数据是：

```
5 279
```

```
7 279
```

则按输出错误处理，不能得分。

【输入输出格式】

输入格式

输入包含 $n+1$ 行：

第 1 行为一个正整数 n ，表示该校参加评选的学生人数。

第 2 到年 $n+1$ 行，每行有 3 个用空格隔开的数字，每个数字都在 0 到 100 之间。第 j 行的 3 个数字依次表示学号为 $j-1$ 的学生的语文、数学、英语的成绩。每个学生的学号按照输入顺序编号为 $1 \sim n$ （恰好是输入数据的行号减 1）。

所给的数据都是正确的，不必检验。

输出格式

输出共有 5 行，每行是两个用空格隔开的正整数，依次表示前 5 名学生的学号和总分。

【输入输出样例】

样例输入 1

```
6
90 67 80
87 66 91
78 89 91
88 99 77
67 89 64
78 89 98
```

样例输出 1

```
6 265
4 264
3 258
2 244
1 237
```

样例输入 2

```
8
80 89 89
88 98 78
90 67 80
87 66 91
78 89 91
88 99 77
67 89 64
78 89 98
```

样例输出 2

```
8 265
2 264
6 264
1 258
5 258
```

【数据使用】

50%的数据满足：各学生的总成绩各不相同

100%的数据满足： $6 \leq n \leq 300$

5 T2 真题

5.1 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：

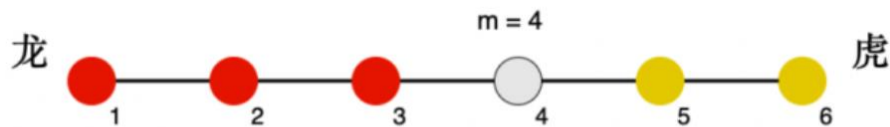


图 2. $n = 6, m = 4$ 的示例

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， $n \leq 10^5, c_i, s_1, s_2 \leq 10^9$

5.2 NOIP2017 图书管理员 (librarian.cpp)

【问题描述】

图书馆中每本书都有一个图书编码，可以用于快速检索图书，这个图书编码是一个正整数。每位借书的读者手中有一个需求码，这个需求码也是一个正整数。如果一本书的图书编码恰好以读者的需求码结尾，那么这本书就是这位读者所需要的。小 D 刚刚当上图书馆的管理员，她知道图书馆里所有书的图书编码，她请你帮她写一个程序，对于每一位读者，求出他所需要的书中图书编码最小的那本书，如果没有他需要的书，请输出-1。

【输入格式】

输入文件名为 librarian.in。输入文件的第一行，包含两个正整数 n 和 q ，以一个空格分开，分别代表图书馆里书的数量和读者的数量。接下来的 n 行，每行包含一个正整数，代表图书馆里某本书的图书编码。接下来的 q 行，每行包含两个正整数，以一个空格分开，第一个正整数代表图书馆里读者的需求码的长度，第二个正整数代表读者的需求码。

【输出格式】

输出文件名为 librarian.out。输出文件有 q 行，每行包含一个整数，如果存在第 i 个读者所需要的书，则在第 i 行输出第 i 个读者所需要的书中图书编码最小的那本书的图书编码，否则输出-1。

【输入输出样例】

librarian.in	librarian.out
5 5	23
2123	1123
1123	-1
23	-1
24	-1
24	
2 23	
3 123	
3 124	
2 12	
2 12	

【输入输出样例说明】

第一位读者需要的书有 2123、1123、23，其中 23 是最小的图书编码。第二位读者需要的书有 2123、1123，其中 1123 是最小的图书编码。对于第三位，第四位和第五位读者，没有书的图书编码以他们的需求码结尾，即没有他们需要的书，输出-1。

【数据规模与约定】

对于 20%的数据， $1 \leq n \leq 2$ 。

另有 20%的数据， $q = 1$ 。

另有 20%的数据，所有读者的需求码的长度均为 1。

另有 20%的数据，所有的图书编码按从小到大的顺序给出。

对于 100%的数据， $1 \leq n \leq 1,000$ ， $1 \leq q \leq 1,000$ ，所有的图书编码和需求码均不超过 10,000,000。

5.3 NOIP2016 日期回文 (date.cpp)

【问题描述】

在日常生活中，通过年、月、日这三个要素可以表示出一个唯一确定的日期。牛牛习惯用 8 位数字表示一个日期，其中，前 4 位代表年份，接下来 2 位代表月份，最后 2 位代表日期。显然：一个日期只有一种表示方法，而两个不同的日期的表示方法不会相同。

牛牛认为，一个日期是回文的，当且仅当表示这个日期的 8 位数字是回文的。现在，牛牛想知道：在他指定的两个日期之间包含这两个日期本身)，有多少个真实存在的日期是回文的。一个 8 位数字是回文的，当且仅当对于所有的 i ($1 \leq i \leq 8$) 从左向右数的第 i 个数字和第 $9-i$ 个数字 (即从右向左数的第 i 个数字) 是相同的。

例如：

- 对于 2016 年 11 月 19 日，用 8 位数字 20161119 表示，它不是回文的。
- 对于 2010 年 1 月 2 日，用 8 位数字 20100102 表示，它是回文的。
- 对于 2010 年 10 月 2 日，用 8 位数字 20101002 表示，它不是回文的。

每一年中都有 12 个月份：其中，1、3、5、7、8、10、12 月每个月有 31 天；4、6、9、11 月每个月有 30 天；而对于 2 月，闰年时有 29 天，平时有 28 天。一个年份是闰年当且仅当它满足下列两种情况其中的一种：

1. 这个年份是 4 的整数倍，但不是 100 的整数倍；
2. 这个年份是 400 的整数倍。

【输入格式】

输入包括两行，每行包括一个 8 位数字。第一行表示牛牛指定的起始日期。第二行表示牛牛指定的终止日期。保证 date_i 和都是真实存在的日期，且年份部分一定为 4 位数字，且首位数字不为 0。保证 date_1 一定不晚于 date_2 。

【输入输出样例】

输出一行，包含一个整数，表示在 date1 和 date2 之间，有多少个日期是回文的。

【样例输入 1】

20110101

20111231

【样例输出 1】

1

【样例输入 2】

20000101

20101231

【样例输出 2】

2

【样例说明】

对于样例 1，符合条件的日期是 20111102。

对于样例 2，符合条件的日期是 20011002 和 20100102。

【子任务】

对于 60%的数据，满足 date1 = date2。

5.4 NOIP2015 扫雷游戏 (mine.cpp)

【问题描述】

扫雷游戏是一款十分经典的单机小游戏。在 n 行 m 列的雷区中有一些格子含有地雷（称之为地雷格），其他格子不含地雷（称之为非地雷格）。玩家翻开一个非地雷格时，该格将会出现一个数字来提示周围格子中有多少个是地雷格。游戏的目标是在不翻出任何地雷格的条件下，找出所有的非地雷格。现在给出 n 行 m 列的雷区中的地雷分布，要求计算出每个非地雷格周围的地雷格数。注：一个格子的周围格子包括其上、下、左、右、左上、右上、左下、右下八个方向上与之直接相邻的格子。

【输入格式】

输入文件名为 mine.in。输入文件第一行是用一个空格隔开的两个整数 n 和 m ，分别表示雷区的行数和列数。接下来 n 行，每行 m 个字符，描述了雷区中的地雷分布情况。字符 '*' 表示相应格子是地雷格，字符 '?' 表示相应格子是非地雷格。相邻字符之间无分隔符。

【输出格式】

输出文件名为 mine.out。输出文件包含 n 行，每行 m 个字符，描述整个雷区。用 '*' 表示地雷格，用周围的地雷个数表示非地雷格。相邻字符之间无分隔符。

【输入输出样例 1】

mine.in	mine.out
3 3	*10
*??	221
???	1*1
?*?	

【输入输出样例 2】

mine.in	mine.out
2 3	2*1
*??	*21
*??	

【数据说明】 对于 100%的数据， $1 \leq n \leq 100$ ， $1 \leq m \leq 100$ 。

5.5 NOIP2014 比例简化 (ratio.cpp)

【问题描述】

在社交媒体上，经常会看到针对某一个观点同意与否的民意调查以及结果。例如，对某一观点表示支持的有 1498 人，反对的有 902 人，那么赞同与反对的比例可以简单的记为 1498:902。不过，如果把调查结果就以这种方式呈现出来，大多数人肯定不会满意。因为这个比例的数值太大，难以一眼看出它们的关系。对于上面这个例子，如果把比例记为 5:3，虽然与真实结果有一定的误差，但依然能够较为准确地反映调查结果，同时也显得比较直观。现给出支持人数 A ，反对人数 B ，以及一个上限 L ，请你将 A 比 B 化简为 A' 比 B' ，要求在 A' 和 B' 均不大于 L 且 A' 和 B' 互质（两个整数的最大公约数是 1）的前提下， $A'/B' \geq A/B$ 且 $A'/B' - A/B$ 的值尽可能小。

【输入格式】

输入共一行，包含三个整数 A ， B ， L ，每两个整数之间用一个空格隔开，分别表示支持人数、反对人数以及上限。

【输出格式】

输出共一行，包含两个整数 A' ， B' ，中间用一个空格隔开，表示化简后的比例。

【输入样例】

1498 902 10

【输出样例】

5 3

【数据说明】

对于 100%的数据, $1 \leq A \leq 1,000,000$, $1 \leq B \leq 100,000,000$, $1 \leq L \leq 100$, $A/B \leq L$

5.6 NOIP2013 表达式求值 (expr.cpp)

【问题描述】

给定一个只包含加法和乘法的算术表达式, 请你编程计算表达式的值。

【输入格式】

输入仅有一行, 为需要你计算的表达式, 表达式中只包含数字、加法运算符“+”和乘法运算符“*”, 且没有括号, 所有参与运算的数字均为 0 到 $2^{31}-1$ 之间的整数。输入数据保证这一行只有 0~9、+、* 这 12 种字符。

【输出格式】

输出只有一行, 包含一个整数, 表示这个表达式的值。注意: 当答案长度多于 4 位时, 请只输出最后 4 位, 前导 0 不输出。

【输入样例】

1+1*3+4

【输出样例】

8

【数据说明】

对于 100%的数据, 运算符 ≤ 1000000

5.7 NOIP2012 寻宝 (treasure.cpp)

【问题描述】

传说很遥远的藏宝楼顶层藏着诱人的宝藏。小明历尽千辛万苦终于找到传说中的这个藏宝楼, 藏宝楼的门口竖着一个木板, 上面写有几个大字: 寻宝说明书。说明书的内容如下: 藏宝楼共有 $N+1$ 层, 最上面一层是顶层, 顶层有一个房间里藏着宝藏。除了顶层外, 藏宝楼另有 N 层, 每层 M 个房间, 这 M 个房间围成一圈并按逆时针方向依次编号为 0, ..., $M-1$ 。其中一些房间有通往上一层的楼梯, 每层楼的楼梯设计可能不同。每个房间里有一个指示牌, 指示牌上有一个数字 x , 表示从这个房间开始按逆时针方向选择第 x 个有楼梯的房间 (假定该房间的编号为 k), 从该房间上楼, 上楼后到达上一层的 k 号房间。比如当前房间的指示牌上写着 2, 则按逆时针方向开始尝试, 找到第 2 个有楼梯的房间, 从该房间上楼。如果当前房间本身就有楼梯通向上一层, 该房间作为第一个有楼梯的房间。寻宝说明书的最后用红色大号字体写着: “寻宝须知: 帮助你找到每层上楼房间的指示牌上的数字 (即每层第一个进入的房间内指示牌上的数字) 总和为打开宝箱的密钥”。请帮助小明算出这个打开宝箱的密钥。

【输入格式】

第一行 2 个整数 N 和 M , 之间用一个空格隔开。 N 表示除了顶层外藏宝楼共 N 层楼, M 表示除顶层外每层楼有 M 个房间。接下来 $N*M$ 行, 每行两个整数, 之间用一个空格隔开, 每行描述一个房间内的情况, 其中第 $(i-1)*M+j$ 行表示第 i 层 $j-1$ 号房间的情况 ($i=1, 2, \dots, N$; $j=1, 2, \dots, M$)。第一个整数表示该房间是否有楼梯通往上一层 (0 表示没有, 1 表示有), 第二个整数表示指示牌上的数字。注意, 从 j 号房间的楼梯爬到上一层到达的房间一定也是 j 号房间。最后一行, 一个整数, 表示小明从藏宝楼底层的几号房间进入开始寻宝 (注: 房间编号从 0 开始)。

【输出格式】

输出只有一行, 一个整数, 表示打开宝箱的密钥, 这个数可能会很大, 请输出对 20123 取模的结果即可。

【输入样例】

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

【输出样例】

5

第一层:

0 号房间, 有楼梯通往上层, 指示牌上的数字是 2;
1 号房间, 无楼梯通往上层, 指示牌上的数字是 3;
2 号房间, 有楼梯通往上层, 指示牌上的数字是 4;

第二层:

0 号房间, 无楼梯通往上层, 指示牌上的数字是 1;

1 号房间, 有楼梯通往上层, 指示牌上的数字是 5;

2 号房间, 有楼梯通往上层, 指示牌上的数字是 2;

小明首先进入第一层(底层)的 1 号房间, 记下指示牌上的数字为 3, 然后从这个房间开始, 沿逆时针方向选择第 3 个有楼梯的房间 2 号房间进入, 上楼后到达第二层的 2 号房间, 记下指示牌上的数字为 2, 由于当前房间本身有楼梯通向上层, 该房间作为第一个有楼梯的房间。因此, 此时沿逆时针方向选择第 2 个有楼梯的房间即为 1 号房间, 进入后上楼梯到达顶层。这时把上述记下的指示牌上的数字加起来, 即 $3+2=5$, 所以打开宝箱的密钥就是 5。

【数据说明】

对于 50%数据, 有 $0 < N \leq 1000$, $0 < x \leq 10000$;

对于 100%数据, 有 $0 < N \leq 10000$, $0 < M \leq 100$, $0 < x \leq 1,000,000$ 。

5.8 NOIP2011 统计单词数 (stat.cpp)

【问题描述】

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】

共 2 行。第 1 行为一个字符串, 其中只含字母, 表示给定单词; 第 2 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。

【输出格式】

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。

【输入输出样例】

输入样例 1:

To

to be or not to be is a question

输出样例 1:

2 0

输入样例 2:

to

Did the Ottoman Empire lose its power at that time

输出样例

-1

【数据说明】

$1 \leq \text{单词长度} \leq 10$ 。

$1 \leq \text{文章长度} \leq 1,000,000$ 。

5.9 NOIP2010 接水问题 (water.cpp)

【问题描述】

学校里有一个水房, 水房里一共装有 m 个龙头可供同学们打开水, 每个龙头每秒钟的供水量相等, 均为 1。现在有 n 名同学准备接水, 他们的初始接水顺序已经确定。将这些同学按接水顺序从 1 到 n 编号, i 号同学的接水量为 w_i 。接水开始时, 1 到 m 号同学各占一个水龙头, 并同时打开水龙头接水。当其中某名同学 j 完成其接水量要求 w_j 后, 下一名排队等候接水的同学 k 马上接替 j 同学的位置开始接水。这个换人的过程是瞬间完成的, 且没有任何水的浪费。即 j 同学第 x 秒结束时完成接水, 则 k 同学第 $x+1$ 秒立刻开始接水。若当前接水人数 n 不足 m , 则只有 n 个龙头供水, 其它 $m-n$ 个龙头关闭。现在给出 n 名同学的接水量, 按照上述接水规则, 问所有同学都接完水需要多少秒。

【输入格式】

第 1 行 2 个整数 n 和 m , 用一个空格隔开, 分别表示接水人数和龙头个数。

第 2 行 n 个整数 w_1, w_2, \dots, w_n , 每两个整数之间用一个空格隔开, w_i 表示 i 号同学的接水量。

【输出格式】

输出只有一行, 1 个整数, 表示接水所需的总时间。

【输入样例】

5 3

4 4 1 2 1

8 4

23 71 87 32 70 93 80 76

【输出样例】

4
163

【数据说明】

【输入输出样例 1 说明】

第 1 秒, 3 人接水。第 1 秒结束时, 1、2、3 号同学每人的已接水量为 1, 3 号同学接完水, 4 号同学接替 3 号同学开始接水。

第 2 秒, 3 人接水。第 2 秒结束时, 1、2 号同学每人的已接水量为 2, 4 号同学的已接水量为 1。第 3 秒, 3 人接水。第 3 秒结束时, 1、2 号同学每人的已接水量为 3, 4 号同学的已接水量为 2。4 号同学接完水, 5 号同学接替 4 号同学开始接水。第 4 秒, 3 人接水。第 4 秒结束时, 1、2 号同学每人的已接水量为 4, 5 号同学的已接水量为 1。1、2、5 号同学接完水, 即所有人完成接水。总接水时间为 4 秒。

$1 \leq n \leq 10000$, $1 \leq m \leq 100$ 且 $m \leq n$;

$1 \leq w_i \leq 100$ 。

5.10 NOIP2009 分数线划定 (sorce.cpp)

【问题描述】

世博会志愿者的选拔工作正在 A 市如火如荼的进行。为了选拔最合适的人才, A 市对所有报名的选手进行了笔试, 笔试分数达到面试分数线的选手方可进入面试。面试分数线根据计划录取人数的 150% 划定, 即如果计划录取 m 名志愿者, 则面试分数线为排名第 $m \times 150\%$ (向下取整) 名的选手的分数, 而最终进入面试的选手为笔试成绩不低于面试分数线的所有选手。现在就请你编写程序划定面试分数线, 并输出所有进入面试的选手的报名号和笔试成绩。

【输入格式】

第一行, 两个整数 n, m ($5 \leq n \leq 5000$, $3 \leq m \leq n$), 中间用一个空格隔开, 其中 n 表示报名参加笔试的选手总数, m 表示计划录取的志愿者人数。输入数据保证 $m \times 150\%$ 向下取整后小于等于 n 。第二行到第 $n+1$ 行, 每行包括两个整数, 中间用一个空格隔开, 分别是选手的报名号 k ($1000 \leq k \leq 9999$) 和该选手的笔试成绩 s ($1 \leq s \leq 100$)。数据保证选手的报名号各不相同。

【输出格式】

第一行, 有两个整数, 用一个空格隔开, 第一个整数表示面试分数线; 第二个整数为进入面试的选手的实际人数。从第二行开始, 每行包含两个整数, 中间用一个空格隔开, 分别表示进入面试的选手的报名号和笔试成绩, 按照笔试成绩从高到低输出, 如果成绩相同, 则按报名号由小到大的顺序输出。

【输入样例】

6 3
1000 90
3239 88
2390 95
7231 84
1005 95
1001 88

【输出样例】

88 5
1005 95
2390 95
1000 90
1001 88
3239 88

【数据说明】

$m \times 150\% = 3 \times 150\% = 4.5$, 向下取整后为 4。保证 4 个人进入面试的分数线为 88, 但因为 88 有重分, 所以所有成绩大于等于 88 的选手都可以进入面试, 故最终有 5 个人进入面试。

5.11 NOIP2008 排座椅 (seat.cpp)

【问题描述】

上课的时候总有一些同学和前后左右的人交头接耳, 这是令小学班主任十分头疼的一件事情。不过, 班主任小雪发现了一些有趣的现象, 当同学们的座次确定下来之后, 只有有限的 D 对同学上课时才会交头接耳。同学们在教室中坐成了 M 行 N 列, 坐在第 i 行第 j 列的同学的位置是 (i, j) , 为了方便同学们进出, 在教室中设置了 K 条横向的通道, L 条纵向的通道。于是, 聪明的小雪想到了一个办法, 或许可以减少上课时学生交头接耳的问题: 她打算重新摆放桌椅, 改变同学们桌椅间通道的位置, 因为如果一条通道隔开了两个会交头接耳的同学, 那么他们就不会交头接耳了。请你帮忙给小雪编写一个程序, 给出最好的通道划分方案。在该方案下, 上课时交头接耳的学生对数最少。

【输入格式】

第一行, 有 5 个用空格隔开的整数, 分别是 M, N, K, L, D ($2 \leq N, M \leq 1000$, $0 \leq K < M$, $0 \leq L < N$, $D \leq 2000$)。接下来

D 行，每行有 4 个用空格隔开的整数，第 i 行的 4 个整数 X_i, Y_i, P_i, Q_i ，表示坐在位置 (X_i, Y_i) 与 (P_i, Q_i) 的两个同学会交头接耳(输入保证他们前后相邻或者左右相邻)。输入数据保证最优方案的唯一性。输入数据保证最优方案的唯一性。

【输出格式】

共两行。

第一行包含 K 个整数， $a_1 a_2 \dots a_K$ ，表示第 a_1 行和第 a_1+1 行之间、第 a_2 行和第 a_2+1 行之间、 \dots 、第 a_K 行和第 a_K+1 行之间要开辟通道，其中 $a_i < a_{i+1}$ ，每两个整数之间用空格隔开（行尾没有空格）。

第二行包含 L 个整数， $b_1 b_2 \dots b_L$ ，表示第 b_1 列和第 b_1+1 列之间、第 b_2 列和第 b_2+1 列之间、 \dots 、第 b_L 列和第 b_L+1 列之间要开辟通道，其中 $b_i < b_{i+1}$ ，每两个整数之间用空格隔开（行尾没有空格）。

【输入样例】

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

【输出样例】

```
2
2 4
```

6 算法挑战

6.1 进制转换 (change.cpp)

【问题描述】

将任意一个 n 进制整数 x 转换成十进制。

【输入格式】

第 1 行 1 个正整数 n ， $1 < n < 10$ ；

第 2 行 1 个整数 x 。

【输出格式】

一行一个数，表示转换得到的十进制数，保证答案不超过 2147483647。

【输入样例】

```
2
100110
```

【输出样例】

```
38
```

6.2 数列 (series.cpp)

【问题描述】

给定一个正整数 k ，把所有 k 的方幂及所有有限个互不相等的 k 的方幂之和构成一个递增的序列。例如，当 $k=3$ 时，这个序列是：1, 3, 4, 9, 10, 12, 13, ...

请求出这个序列的第 n 项的值(用十进制数表示)。

【输入格式】

一行两个正整数 k 和 n ，之间用一个空格隔开，且 $3 \leq k \leq 15$ ， $10 \leq n \leq 1000$ 。

【输出格式】

一行一个正整数。

【输入样例】

```
3 100
```

【输出样例】

```
981
```

6.3 高精度乘法 (mult.cpp)

【问题描述】

输入两个 100 位以内的正整数，输出它们的乘积。

【输入样例】

```
123456789
987654321
```

【输出样例】

```
121932631112635269
```

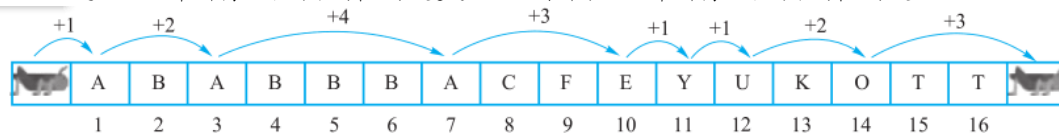
6.4 蚱蜢 (grasshopper.cpp)

【问题描述】

有一天，一只蚱蜢像往常一样在草地上愉快地跳跃，它发现了一条写满了英文字母的纸带。

蚱蜢只能在元音字母(A、E、I、O、U、Y)间跳跃，一次跳跃所需的能力是两个位置的差。纸带所需的能力值为蚱蜢从纸

带开头的 前一个位置根据规则跳到纸带结尾的后一个位置的过程中能力的最大值。
 蚱蜢想知道跳跃纸带所需的能力值(最小)是多少。如图所示的纸带所需的能力值(最小)是 4。



【输入格式】

一行一个字符串，字符串长不超过 100。

【输出格式】

一行一个整数，代表(最小)能力值。

【输入样例】

```
KMLPTGFHNBVCDFGHNMBVXWSQFDCVBNHTJKLPMNFVCKMLPTGFHNBVCDF-
GHNMBVXWSQFDCVBNHTJKLPMNFVC
```

【输出样例】

85

6.5 遭遇战 (battle.cpp)

【问题描述】

小林和小华在一个 $n \times n$ 的矩形方格里玩游戏，矩形左上角为 $(0, 0)$ ，右下角为 $(n-1, n-1)$ 。

两人同时进入地图的随机位置，并以相同速度进行走位。为了隐蔽性，两人都不会再走自己走过的格子。如果两人向某一方向前进，那么他们会跑到不能跑为止，当不能跑的时候，小林会向右转，小华则会向左转，如果不能跑，则不再动。现在已知两人进入地图的初始位置和方向，请算出两人遭遇的位置。

【输入格式】

第 1 行 1 个正整数 t ，表示测试数据组数， $1 \leq t \leq 10$ 。

接下来的 t 组数据，每组数据的第 1 行包含 1 个整数 n ， $1 \leq n \leq 1000$ 。

第 2 行包含 3 个整数 x 、 y 和 d ，表示小林的初始位置和一开始跑的方向。其中， $d=0$ 表示东； $d=1$ 表示南； $d=2$ 表示西； $d=3$ 表示北。

第 3 行与第 2 行格式相同，但描述的是小华。

【输出格式】

输出 t 行，若会遭遇，则包含两个整数，表示他们第一次相遇格子的坐标，否则输出“-1”。

第 2 行包含 3 个整数 x 、 y 和 d ，表示小林的初始位置和一开始跑的方向。其中， $d=0$ 表示东； $d=1$ 表示南； $d=2$ 表示西； $d=3$ 表示北。

第 3 行与第 2 行格式相同，但描述的是小华。

【输出格式】

输出 t 行，若会遭遇，则包含两个整数，表示他们第一次相遇格子的坐标，否则输出“-1”。

【输入样例】

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

【输出样例】

```
-1
1 3
```

6.6 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）。比如有这么一份记录，（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

输出格式:

【输入输出样例】

WWWWWWWWWWWWWWWWWWWWWWWWWWWWWW

WWLWE

11:0

11:0

1:1

21:0

2:1

每行至多 25 个字母, 最多有 2500 行

【问题描述】

【输入格式】

【输出格式】

【输入样例】

2

【输出样例】

3

【问题描述】

【输入格式】

接下来的 n 行, 每行 2 个数 x_i 和 y_i , 表示城市 i 的坐标。

【输出格式】

【输入样例】

4

0 0

83

11 -1

10 0

【输出样例】

14

【样例说明】

【数据规模】

对于 40% 的数据满足: $n \leq 1000$ 。

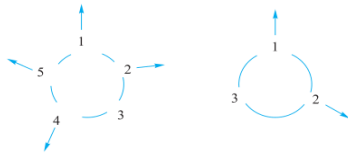
对于 100% 的数据满足： $3 \leq n \leq 100000$, $-1000 \leq x_i, y_i \leq 1000$ 。

【问题描述】

第 24 页共 36 页

地离开，其余剩下的人将重复以上过程，比幸存者号码高的人每人将得到 1 块钱后离开。一旦经过这样的过程后，人数不再减少，最后剩下的那些人将得到 2 块钱。请计算一下组织者一共要付出多少钱？

如图所示，第一轮有 5 人，幸存者是 3，所以 4、5 得到 1 块钱后离开，下一轮幸存者仍然是 3，因此没有人离开，所以每人得到 2 块钱，总共要付出 $2+2\times 3=8$ 块钱。



【输入格式】

一行一个整数 n ，不超过 32767。

【输出格式】

一行一个整数，不超过 65535，表示总共要付出多少钱。

【输入样例】

10

【输出样例】

13

6.10 找伪币 (coin.cpp)

【问题描述】

给出 16 个一模一样的硬币，其中有 1 个是伪造的，并且那个伪造的硬币比真的硬币要轻一些，本题的任务是找出这个伪造的硬币。为了完成这一任务，将提供一台可用来比较两组硬币重量的仪器，利用这台仪器，可以知道两组硬币孰轻孰重。

6.11 循环比赛日程表 (schedule.cpp)

【问题描述】

设有 n 位选手的循环比赛，其中 $n=2^m$ ， m 为正整数。要求每位选手要与其他 $n-1$ 位选手都赛一次。每名选手每天比赛一次，循环赛共进行 $n-1$ 天，要求每天没有选手轮空。下图是 8 位选手时 ($m=3$) 的循环比赛表，表中第一行为 8 位选手的编号，下面 7 行，依次是每位选手每天的对手。

1	2	3	4	5	6	7	8
2	1	4	3	6	5	8	7
3	4	1	2	7	8	5	6
4	3	2	1	8	7	6	5
5	6	7	8	1	2	3	4
6	5	8	7	2	1	4	3
7	8	5	6	3	4	1	2
8	7	6	5	4	3	2	1

6.12 取模运算 (model.cpp)

【问题描述】

定义“取模”运算：对于正整数 a 和 p ， $a \% p$ 表示 a 除以 p 的余数，又称“模”运算。现在，输入三个正整数 b 、 p 、 k ，请编程计算 $b^p \% k$ 的值。

【输入格式】

一行三个正整数，分别表示 b 、 p 、 k 的值。其中， b 、 p 、 $k \times k \leq 2147483647$ 。

【输出格式】

一行一个整数，表示 $b^p \% k$ 的值。

【输入样例】

2 10 9

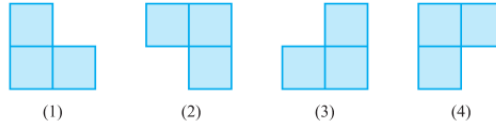
【输出样例】

7

6.13 地毯填铺 (carpet.cpp)

【问题描述】

相传在一个古老的阿拉伯国家有一座宫殿，宫殿里有个四四方方的格子迷宫。国王选驸马的方法非常特殊，也非常简单：公主就站在其中一个方格子上，只要谁能用地毯将除公主所站立的格子之外的所有地方盖上，就可以娶到美丽漂亮的公主。毯子的形状只有如图 9.5-2 所示的 4 种，并且每一方格只能用一层地毯，格子迷宫的大小为 $2k$ 的正方形。



【输入格式】

第 1 行 1 个正整数 k , $1 \leq k \leq 10$;

第 2 行 2 个整数 x 和 y , 表示公主所在方格的坐标 (x 为行坐标, y 为列坐标), x 和 y 之间有一个空格。

【输出格式】

输出将迷宫填补完整的方案。

每一补(行)为 $x y c$, 其中 x 和 y 为毯子拐角的行坐标和列坐标, c 为所使用毯子的形状, 具体标号如图 9.5-2 所示, 毯子形状分别用 1、2、3、4 表示, x 、 y 、 c 之间用一个空格隔开。

【输入样例】

3

3 3

【输出样例】

5 5 1

2 2 4

1 1 4

1 4 3

4 1 2

4 4 1

2 7 3

1 5 4

1 8 3

3 6 3

4 8 1

7 2 2

5 1 4

6 3 2

8 1 2

8 4 1

7 7 1

6 6 1

5 8 3

8 5 2

8 8 1

6.14 记忆化斐波那契数列 (fibo.cpp)

【问题描述】

输入 n , $1 \leq n \leq 1000$, 输出斐波那契数列第 n 项模 9997 的值。

【输入样例】

10

【输出样例】

55

6.15 砍伐树木 (tree.cpp)

【问题描述】

小华被大林叫去砍树, 他需要砍倒 m 米长的木材。现在, 小华弄到了一个奇怪的伐木机。伐木机工作过程如下: 小华设置一个高度参数 h (米), 伐木机升起一个巨大的锯片到高度 h , 并锯掉所有的树比 h 高的部分 (当然, 树木不高于 h 米的部分保持不变)。小华就得到树木被锯下的部分。

例如, 如果一行树的高度分别为 20、15、10 和 17 米, 小华把锯片升到 15 米的高度, 切割后树木剩下的高度将是 15、15、10 和 15 米, 而小华将从第 1 棵树得到 5 米, 从第 4 棵树得到 2 米, 共得到 7 米木材。

小华非常关注生态保护, 所以他不会砍掉过多的木材。这正是他为什么要尽可能高地设定伐木机锯片的原因。帮助小华找到伐木机锯片的最大的整数高度 h , 使得他能得到的木材至少为 m 米。换句话说, 如果再升高 1 米, 则他将得不到 m 米木材。

【输入格式】

第 1 行 2 个整数 n 和 m , n 表示树木的数量, m 表示需要的木材总长度。

第 2 行 n 个整数, 表示每棵树的高度, 值均不超过 10 9。保证所有木材长度之和大于 m , 因此必然有解。

【输出格式】

一行一个整数，表示砍树的最高高度。

【输入样例】

```
5 20
4 42 40 26 46
```

【输出样例】

```
36
```

【数据规模】

对于 30% 的数据满足： $1 \leq n \leq 10$ ， $1 \leq m \leq 30$ 。

对于 70% 的数据满足： $1 \leq n \leq 103$ ， $1 \leq m \leq 104$ 。

对于 100% 的数据满足： $1 \leq n \leq 106$ ， $1 \leq m \leq 2 \times 109$ 。

6.16 独木舟 (canoe.cpp)

【问题描述】

旅行社计划组织一个独木舟旅行。租用的独木舟都是一样的，最多乘两人，而且载重有一个限度。现在要节约费用，所以要尽可能地租用最少的舟。本题的任务是读入独木舟的载重量，参加旅行的人数以及每个人的体重，计算出所需要的独木舟数目。

【输入格式】

第 1 行是 w ($80 \leq w \leq 200$)，表示每条独木舟最大的载重量。

第 2 行是正整数 n ($1 \leq n \leq 30000$)，表示参加旅行的人数。

接下来的 n 行，每行是一个正整数 t_i ($5 \leq t_i \leq w$)，表示每个人的重量。

【输出格式】

输出一行一个数，表示最少的独木舟数目。

【输入样例】

```
100
9
90
20
20
30
50
60
70
80
90
```

【输出样例】

```
6
```

6.17 最大整数 (maxInt.cpp)

【问题描述】

设有 n ($n \leq 20$) 个正整数 (小于或等于 2147483647)，将它们连接成一排，组成一个最大的多位整数。例如 $n=3$ ，3 个整数为 13、312 和 343，连接成的最大整数为 34331213。

【输入格式】

第 1 行 1 个整数 n 。

第 2 行为 n 个正整数，每两个数之间用一个空格隔开。

【输出格式】

一行一个数，表示连接成的最大整数。

【输入样例】

```
4
7 13 4 246
```

【输出样例】

```
7424613
```

6.18 排队接水 (water.cpp)

【问题描述】

有 n 个人在一个水龙头前排队接水，假如每个人接水的时间为 T_i ，请编程找出一种这 n 个人排队的顺序，使得 n 个人的平均等待时间最小。

【输入格式】

第 1 行为一个整数 n ；

第 2 行分别表示每人的接水时间 T_1, T_2, \dots, T_n ，每两个数据之间有 1 个空格。

【输出格式】

第 1 行为一种排队顺序，即 $1 \sim n$ 的一种排列，每两个数据之间有 1 个空格。

第 2 行为这种排列方案下的平均等待时间（输出结果精确到小数点后两位）。

【输入样例】

```
10
56 12 1 99 1000 234 33 55 99 812
```

【输出样例】

```
3 2 7 8 1 4 9 6 10 5
291.90
```

6.19 楼层编号 (floor.cpp)

【问题描述】

小林在 NOIP 比赛期间住在“新世界”酒店。和其他酒店不一样的是，这个酒店每天都有一个高能数字 t ，这个数字在楼层中是不会出现的，以 $t=3$ 为例，则 3、13、31、33 等楼层是不存在的，楼层编号为 1, 2, 4, 5, ... 所以实际上的 4 楼才是 3 楼。

已知小林预订了编号为 m 层的房间，并且当天高能数字是 t ，现在他想知道房间所在的真实楼层是多少。

【输入格式】

一行两个整数 m 和 t ， $1 \leq m \leq 100000$ ， $0 \leq t \leq 9$ ，保证 m 对 t 合法。

【输出格式】

一行一个整数，表示真实楼层。

【输入样例】

```
14 3
```

【输出样例】

```
12
```

6.20 最大黑色区域 (area.cpp)

【问题描述】

二值图像是由黑和白两种像素组成的矩形点阵，图像识别的一个操作是求出图像中最大黑区域的面积。请设计一个程序完成二值图像的这个操作。黑区域由黑像素组成，一个黑区域中的每像素至少与该区域中的另一像素相邻，规定一个像素仅与其上、下、左、右的像素相邻。两个不同的黑区域没有相邻的像素。一个黑区域的面积是其所包含的像素数。

【输入格式】

第 1 行含两个整数 n 和 m ， $1 \leq n, m \leq 100$ ，分别表示二值图像的行数与列数。后面 n 行，每行含 m 个整数 0 或 1，其中第 i 行表示图像的第 i 行的 m 个像素，0 表示白像素，1 表示黑像素。每行中的两个数之间用一个空格分隔。

【输出格式】

输出一行一个数，表示相应的图像中最大黑区域的面积。

【输入样例】

```
5 6
0 1 1 0 0 1
1 1 0 1 0 1
0 1 0 0 1 0
0 0 0 1 1 1
1 0 1 1 1 0
```

【输出样例】

```
7
```

6.21 关系网络 (relationship.cpp)

【问题描述】

有 n 个人，他们的编号为 $1 \sim n$ ，其中有一些人相互认识，现在 x 想要认识 y ，可以通过他所认识的人来认识更多的人（如果 a 认识 b ， b 认识 c ，那么 a 可以通过 b 来认识 c ），求出 x 最少需要通过多少人才能认识 y 。

【输入格式】

第 1 行 3 个整数 n, x, y ， $2 \leq n \leq 100$ ；

接下来的 n 行是一个 $n \times n$ 的邻接矩阵， $a[i][j]=1$ 表示 i 认识 j ， $a[i][j]=0$ 表示不认识。

保证 $i=j$ 时， $a[i][j]=0$ ，并且 $a[i][j]=a[j][i]$ 。

【输出格式】

一行一个整数，表示 x 认识 y 最少需要通过的人数。数据保证 x 一定能认识 y 。

【输入样例】

```
5 1 5
0 1 0 0 0
```

```
1 0 1 1 0
0 1 0 1 0
0 1 1 0 1
0 0 0 1 0
```

【输出样例】

2

6.22 溶液模拟器 (simulator.cpp)

【问题描述】

小谢虽然有很多溶液，但是还是没有办法配成想要的溶液，因为万一倒错了就没有办法挽回了。因此，小谢到网上下载了一个溶液配置模拟器。模拟器在计算机中构造一种虚拟溶液，然后可以虚拟地向当前虚拟溶液中加入一定浓度、一定体积的这种溶液，模拟器会快速地算出倒入后虚拟溶液的浓度和体积。当然，如果倒错了可以撤销。

模拟器的使用步骤如下：

1) 为模拟器设置一个初始体积和浓度 V_0 、 $C_0\%$ 。

2) 进行一系列操作，模拟器支持两种操作：

$P(v, c)$ 操作：表示向当前的虚拟溶液中加入体积为 v 浓度为 c 的溶液；

Z 操作：撤销上一步的 P 操作。

【输入格式】

第一行两个整数，表示 V_0 和 C_0 ， $0 \leq C_0 \leq 100$ ；

第二行一个整数 n ，表示操作数， $n \leq 10000$ ；

接下来 n 行，每行一条操作，格式为： P_v_c 或 Z 。

其中 $_$ 代表一个空格，当只剩初始溶液的时候，再撤销就没有用了。

任意时刻质量不会超过 $2^{31}-1$ 。

【输出格式】

n 行，每行两个数 V_i, C_i ，其中 V_i 为整数， C_i 为实数（保留 5 位小数）。

其中，第 i 行表示第 i 次操作以后的溶液体积和浓度。

【输入样例】

```
100 100
2
P 100 0
Z
```

【输出样例】

```
200 50.00000
100 100.00000
```

6.23 背包问题 (pack.cpp)

【问题描述】

小明就要去春游了。妈妈给他买了很多好吃的。小明想把这些吃的都放进他的书包，但他很快发现，妈妈买的东西实在太多了，他必须放弃一些，但又希望能带尽可能多的好吃的。举算法解决一些实际问题。

已知小明的书包最多可以装入总重量为 s 的物品，同时也知道小明妈妈给他买的每样东西的重量。请从这些好吃的中选出若干装入小明的书包中，使得装入物品的总重量正好为 s 。找到任意一组解输出即可。

【输入数据】

第 1 行包含两个正整数 n ($1 \leq n \leq 100$) 和 s ($1 \leq s \leq 10\,000$)，分别代表有 n 件物品和书包的最大承重 s ；

第 2 行包含 n 个正整数，代表每件物品的重量 W_i ($1 \leq W_i \leq 1\,000$)。同行的两个数字之间用一个空格隔开。

【输出数据】

一行包含有若干用一个空格隔开的正整数，代表被放入书包的若干物品各自的重量。若无可行解，则输出 “No Answer !”。

【输入样例 1】

```
8 14
1 3 2 5 9 4 7 6
```

【输出样例 1】

```
1 3 4 6
```

【输入样例 2】

```
3 12
2 8 5
```

【输出样例 2】

```
No Answer!
```

6.24 数的拆分 (number.cpp)

【问题描述】

输入 n ，输出将 n 拆分成若干正整数和的所有方案，即 $n = S_1 + S_2 + \dots + S_k$ 的形式，且 $S_1 \leq S_2 \leq \dots \leq S_k$ ， $n \leq 20$ ，请按照字典序输出。

【输入格式】

一行一个整数 n 。

【输出格式】

所有拆分方案，具体格式参见输出样例。

【输入样例】

4

【输出样例】

1+1+1+1

1+1+2

1+3

2+2

4

total=5

6.25 导弹拦截 (missile.cpp)

【问题描述】

某国为了防御敌国的导弹袭击，研究出一种导弹拦截系统。但是这种导弹拦截系统有一个缺陷：虽然它的第一发炮弹能够到达任意高度，但是以后每一发炮弹都不能高于前一发的高度。

某天，雷达捕捉到敌国的导弹来袭。由于该系统还在试用阶段，所以只有一套系统，因此有可能不能拦截所有的导弹。

输入导弹的枚数和导弹依次飞来的高度，计算这套系统最多能拦截多少导弹，以及如果要拦截所有导弹最少要配备多少套这种导弹拦截系统。

【输入格式】

第 1 行为正整数 n ，表示导弹的枚数。

第 2 行为 n 个正整数，表示依次飞来的导弹高度（不大于 30000，单位：米），每两个数据之间有一个空格。

【输出格式】

第 1 行 1 个正整数，表示一套系统最多能拦截多少导弹。

第 2 行 1 个正整数，表示要拦截所有导弹最少要配备多少套拦截系统。

【输入样例】

8

389 207 155 300 299 170 158 65

【输出样例】

6

2

7 T3 真题

7.1 NOIP2018 摆渡车 (bus.cpp)

【题目描述】

有 n 名同学要乘坐摆渡车从人大附中前往人民大学，第 i 位同学在第 t_i 分钟去等车。只有一辆摆渡车在工作，但摆渡车容量可以视为无限大。摆渡车从人大附中出发、把车上的同学送到人民大学、再回到人大附中（去接其他同学），这样往返一趟总共花费 m 分钟（同学上下车时间忽略不计）。摆渡车要将所有同学都送到人民大学。

凯凯很好奇，如果他能任意安排摆渡车出发的时间，那么这些同学的等车时间之和最小为多少呢？

注意：摆渡车回到人大附中后可以即刻出发。

【输入】

第一行包含两个正整数 n, m ，以一个空格分开，分别代表等车人数和摆渡车往返一趟的时间。

第二行包含 n 个正整数，相邻两数之间以一个空格分隔，第 i 个非负整数 t_i 代表第 i 个同学到达车站的时刻。

【输出】

输出一行，一个整数，表示所有同学等车时间之和的最小值（单位：分钟）。

【输入样例 1】

5 1

3 4 4 3 5

【输入样例 2】

5 5
11 13 1 5 5

【输出样例 1】

0

【输出样例 2】

4

【输入输出样例 1 说明】

同学 1 和同学 4 在第 3 分钟开始等车，等待 0 分钟，在第 3 分钟乘坐摆渡车出发。摆渡车在第 4 分钟回到人大附中。

同学 2 和同学 3 在第 4 分钟开始等车，等待 0 分钟，在第 4 分钟乘坐摆渡车出发。摆渡车在第 5 分钟回到人大附中。

同学 5 在第 5 分钟开始等车，等待 0 分。

【输入输出样例 2 说明】

同学 3 在第 1 分钟开始等车，等待 0 分钟，在第 1 分钟乘坐摆渡车出发。摆渡车在第 6 分钟回到人大附中。

同学 4 和同学 5 在第 5 分钟开始等车，等待 1 分钟，在第 6 分钟乘坐摆渡车出发。摆渡车在第 11 分钟回到人大附中。

同学 1 在第 11 分钟开始等车，等待 2 分钟；同学 2 在第 13 分钟开始等车，等待 0 分钟。他/她们在第 13 分钟乘坐摆渡车出发。自此所有同学都被送到人民大学。总等待时间为 4。可以证明，没有总等待时间小于 4 的方案。

【数据规模与约定】

对于 10% 的数据， $n \leq 10, m = 1, 0 \leq t_i \leq 100$ 。

对于 30% 的数据， $n \leq 20, m \leq 2, 0 \leq t_i \leq 100$ 。

对于 50% 的数据， $n \leq 500, m \leq 100, 0 \leq t_i \leq 10^4$ 。

另有 20% 的数据， $n \leq 500, m \leq 10, 0 \leq t_i \leq 4 \times 10^6$ 。

对于 100% 的数据， $n \leq 500, m \leq 100, 0 \leq t_i \leq 4 \times 10^6$ 。

7.2 NOIP2017 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。

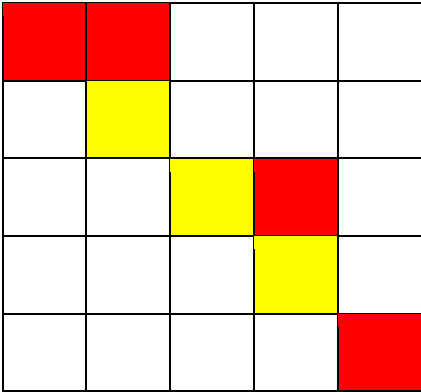
输出一行，一个整数，表示花费的金币的最小值，如果无法到达，输出 -1。

【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】

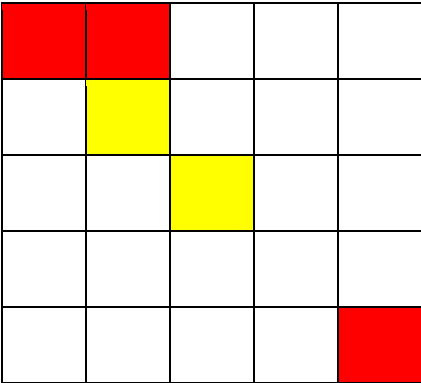


从（1，1）开始，走到（1，2）不花费金币从（1，2）向下走到（2，2）花费 1 枚金币从（2，2）施展魔法，将（2，3）变为黄色，花费 2 枚金币从（2，2）走到（2，3）不花费金币从（2，3）走到（3，3）不花费金币从（3，3）走到（3，4）花费 1 枚金币从（3，4）走到（4，4）花费 1 枚金币从（4，4）施展魔法，将（4，5）变为黄色，花费 2 枚金币，从（4，4）走到（4，5）不花费金币从（4，5）走到（5，5）花费 1 枚金币共花费 8 枚金币。

【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, $1 \leq m \leq 100, 1 \leq n \leq 1,000$ 。

7.3 NOIP2016 海港 (port.cpp)

【问题描述】

小 K 是一个海港的海关工作人员, 每天都有许多船只到达海港, 船上通常有很多来自不同国家的乘客。小 K 对这些到达海港的船只非常感兴趣, 他按照时间记录下了到达海港的每一艘船只情况; 对于第 i 艘到达的船, 他记录了这艘船到达的时间 t_i (单位: 秒), 船上的乘客数量 k_i , 以及每名乘客的国籍 $x(i, 1), x(i, 2), \dots, x(i, k)$ 。小 K 统计了 n 艘船的信息, 希望你帮忙计算出以每一艘船到达时间为止的 24 小时 (24 小时=86400 秒) 内所有乘船到达的乘客来自多少个不同的国家。

形式化地讲, 你需要计算 n 条信息。对于输出的第 i 条信息, 你需要统计满足 $t_i - 86400 < t_p \leq t_i$ 的船只 p , 在所有的 $x(p, j)$ 中, 总共有多少个不同的数。

【输入输出格式】

输入格式:

第一行输入一个正整数 n , 表示小 K 统计了 n 艘船的信息。接下来 n 行, 每行描述一艘船的信息: 前两个整数 t_i 和 k_i 分别表示这艘船到达海港的时间和船上的乘客数量, 接下来 k_i 个整数 $x(i, j)$ 表示船上乘客的国籍。保证输入的 t_i 是递增的, 单位是秒; 表示从小 K 第一次上班开始计时, 这艘船在第 t_i 秒到达海港。保证 $1 \leq n \leq 10^5$, $\sum k_i \leq 3 \times 10^5$, $1 \leq x(i, j) \leq 10^5$, $1 \leq t(i-1) \leq t_i \leq 10^9$ 。其中 $\sum k_i$ 表示所有的 k_i 的和。

输出格式:

输出 n 行, 第 i 行输出一个整数表示第 i 艘船到达后的统计信息。

【输入输出样例】

输入样例#1:

```
3
1 4 4 1 2 2
2 2 2 3
10 1 3
```

输出样例#1:

```
3
4
4
```

输入样例#2:

```
4
1 4 1 2 2 3
3 2 2 3
86401 2 3 4
86402 1 5
```

输出样例#2:

```
3
3
3
4
```

【样例解释 1】

第一艘船在第 1 秒到达海港, 最近 24 小时到达的船是第一艘船, 共有 4 个乘客, 分别是来自国家 4, 1, 2, 2, 共来自 3 个不同的国家;

第二艘船在第 2 秒到达海港, 最近 24 小时到达的船是第一艘船和第二艘船, 共有 $4 + 2 = 6$ 个乘客, 分别是来自国家 4, 1, 2, 2, 2, 3, 共来自 4 个不同的国家;

第三艘船在第 10 秒到达海港, 最近 24 小时到达的船是第一艘船、第二艘船和第三艘船, 共有 $4 + 2 + 1 = 7$ 个乘客, 分别是来自国家 4, 1, 2, 2, 2, 3, 3, 共来自 4 个不同的国家。

【样例解释 2】

第一艘船在第 1 秒到达海港, 最近 24 小时到达的船是第一艘船, 共有 4 个乘客, 分别是来自国家 1, 2, 2, 3, 共来自 3 个不同的国家。

第二艘船在第 3 秒到达海港, 最近 24 小时到达的船是第一艘船和第二艘船, 共有 $4 + 2 = 6$ 个乘客, 分别是来自国家 1, 2, 2, 3, 2, 3, 共来自 3 个不同的国家。

第三艘船在第 86401 秒到达海港, 最近 24 小时到达的船是第二艘船和第三艘船, 共有 $2 + 2 = 4$ 个乘客, 分别是来自国家 2, 3, 3, 4, 共来自 3 个不同的国家。

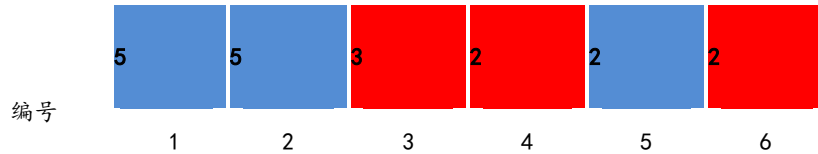
第四艘船在第 86402 秒到达海港, 最近 24 小时到达的船是第二艘船、第三艘船和第四艘船, 共有 $2 + 2 + 1 = 5$ 个乘客,

分别是来自国家 2, 3, 3, 4, 5, 共来自 4 个不同的国家。

7.4 NOIP2015 求和 (sum.cpp)

【问题描述】

一条狭长的纸带被均匀划分出了 n 个格子，格子编号从 1 到 n 。每个格子上都染了一种颜色 $color_i$ (用 $[1, m]$ 中的一个整数表示)，并且写了一个数字 $number_i$ 。



定义一种特殊的三元组: (x, y, z) ，其中 x, y, z 都代表纸带上格子的编号，这里的三元组要求满足以下两个条件：

1. x, y, z 都是整数， $x < y < z$ ， $y - x = z - y$
2. $color_x = color_z$

满足上述条件的三元组的分数规定为 $(x + z) * (number_x + number_z)$ 。整个纸带的分数规定为所有满足条件的三元组的分数的和。这个分数可能会很大，你只要输出整个纸带的分数除以 10,007 所得的余数即可。

【输入格式】

输入文件名为 sum.in。

第一行是用一个空格隔开的两个正整数 n 和 m ， n 代表纸带上格子的个数， m 代表纸带上颜色的种类数。

第二行有 n 个用空格隔开的正整数，第 i 个数字 $number_i$ 代表纸带上编号为 i 的格子上面写的数字。

第三行有 n 个用空格隔开的正整数，第 i 个数字 $color_i$ 代表纸带上编号为 i 的格子染的颜色。

【输出格式】

输出文件名为 sum.out。 共一行，一个整数，表示所求的纸带分数除以 10,007 所得的余数。

【输入输出样例 1】

sum.in	sum.out
6 2 5 5 3 2 2 2 2 2 1 1 2 1	82

【输入输出样例 1 说明】

纸带如题目描述中的图所示。

所有满足条件的三元组为: $(1, 3, 5)$ ， $(4, 5, 6)$ 。

所以纸带的分数为 $(1 + 5) * (5 + 2) + (4 + 6) * (2 + 2) = 42 + 40 = 82$ 。

7.5 NOIP2014 螺旋矩阵 (spiral.cpp)

【问题描述】

一个 n 行 n 列的螺旋矩阵可由如下方法生成:从矩阵的左上角(第 1 行第 1 列)出发,初始时向右移动;如果前方是未曾经过的格子,则继续前进,否则右转;重复上述操作直至经过矩阵中所有格子。根据经过顺序,在格子中依次填入 1, 2, 3, ..., n^2 , 便构成了一个螺旋矩阵。

下图是一个 $n = 4$ 时的螺旋矩阵。

1	2	3	4
12	13	14	5
11	16	15	6
10	9	8	7

现给出矩阵大小 n 以及 i 和 j , 请你求出该矩阵中第 i 行第 j 列的数是多少。

【输入】

输入共一行, 包含三个整数 n, i, j , 每两个整数之间用一个空格隔开, 分别表示矩阵大小、待求的数所在的行号和列号。

【输出】

输出共一行, 包含一个整数, 表示相应矩阵中第 i 行第 j 列的数。

【输入样例】

4 2 3

【输出样例】

14

【数据说明】

对于 50%的数据, $1 \leq n \leq 100$;

对于 100%的数据, $1 \leq n \leq 30,000, 1 \leq i \leq n, 1 \leq j \leq n$ 。

【输入输出样例 2】

sum. in	sum. out
15 4 5 10 8 2 2 2 9 9 7 7 5 6 4 2 4 2 2 3 3 4 3 3 2 4 4 4 4 1 1 1	1388

【输入输出样例 3】见选手目录下的 sum/sum3.in 和 sum/sum3.ans。

【数据说明】

- 对于第 1 组至第 2 组数据， $1 \leq n \leq 100$ ， $1 \leq m \leq 5$ ；
- 对于第 3 组至第 4 组数据， $1 \leq n \leq 3000$ ， $1 \leq m \leq 100$ ；
- 对于第 5 组至第 6 组数据， $1 \leq n \leq 100000$ ， $1 \leq m \leq 100000$ ，且不存在出现次数超过 20 的颜色；
- 对于全部 10 组数据， $1 \leq n \leq 100000$ ， $1 \leq m \leq 100000$ ， $1 \leq color_i \leq m$ ， $1 \leq number_i \leq 100000$ 。