入门组模拟卷

题目名称	自幂数判断	地图探险	学习小组	团队协作
题目类型	传统型	传统型	传统型	传统型
目录	number	explore	group	teamwork
可执行文件名	number	explore	group	teamwork
输入文件名	number.in	explore.in	group.in	teamwork.in
输出文件名	number.out	explore.out	group.out	teamwork.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	number.cpp	explore.cpp	group.cpp	teamwork.cpp
对于 C 语言	number.c	explore.c	group.c	teamwork.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，申述时一律不予受理。
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。

自幂数判断（number）

【题目描述】

自幂数是指，一个 N 位数，满足各位数字的某个整数次方之和是本身。例如，153 是 3 位数，其每位数的 3 次方之和， $1^3+5^3+3^3=153$，因此 153 是自幂数；1634 是 4 位数，其每位数的 4 次方之和，$1^4+6^4+3^4+4^4=1634$，因此 1634 是自幂数。现在，输入若干个正整数，请判断它们是否是自幂数。

【输入格式】

从文件 **number.in** 中读入数据。

输入第一行是一个正整数 M，表示有 M 个待判断的正整数。约定 $1 \le M \le 100$。从第 2 行开始的 M 行，每行一个待判断的正整数。约定这些正整数均小于 10^8。

【输出格式】

输出到文件 **number.out** 中。

输出 M 行，如果对应的待判断正整数为自幂数，则输出英文大写字母 T，否则输出英文大写字母 F。

【样例 1 输入】

```
3
152
111
153
```

【样例 1 输出】

```
F
F
T
```

地图探险（explore）

【题目描述】

小 A 打算前往一片丛林去探险。丛林的地理环境十分复杂，为了防止迷路，他先派遣了一个机器人前去探路。

丛林的地图可以用一个 n 行 m 列的字符表来表示。我们将第 i 行第 j 列的位置的坐标记作 $(i,j)(1 \le i \le n, 1 \le j \le m)$。如果这个位置的字符为 `x`，即代表这个位置上有障碍，不可通过。反之，若这个位置的字符为 `.`，即代表这个位置是一片空地，可以通过。

这个机器人的状态由位置和朝向两部分组成。其中位置由坐标 $(x,y)(1 \le x \le n, 1 \le y \le m)$ 刻画，它表示机器人处在地图上第 x 行第 y 列的位置。而朝向用一个 $0 \sim 3$ 的整数 d 表示，其中 $d=0$ 代表向东，$d=1$ 代表向南，$d=2$ 代表向西，$d=3$ 代表向北。

初始时，机器人的位置为 (x_0, y_0)，朝向为 d_0。**保证初始时机器人所在的位置为空地**。接下来机器人将要进行 k 次操作。每一步，机器人将按照如下的模式操作：

1. 假设机器人当前处在的位置为 (x,y)，朝向为 d。则它的方向上的下一步的位置 (x',y') 定义如下：若 $d=0$，则令 $(x',y')=(x,y+1)$，若 $d=1$，则令 $(x',y')=(x+1,y)$，若 $d=2$，则令 $(x',y')=(x,y-1)$，若 $d=3$，则令 $(x',y')=(x-1,y)$。
2. 接下来，机器人判断它下一步的位置是否在地图内，且是否为空地。具体地说，它判断 (x',y') 是否满足 $1 \le x' \le n, 1 \le y' \le m$，且 (x',y') 位置上是空地。如果条件成立，则机器人会向前走一步。它新的位置变为 (x',y')，且朝向不变。如果条件不成立，则它会执行“向右转”操作。也就是说，令 $d'=(d+1) \bmod 4$（即 $d+1$ 除以 4 的余数），且它所处的位置保持不变，但朝向由 d 变为 d'。

小 A 想要知道，在机器人执行完 k 步操作之后，地图上所有被机器人经过的位置（包括起始位置）有几个。

【输入格式】

从文件 ***explore.in*** 中读入数据。

本题有多组测试数据。

输入的第一行包含一个正整数 T，表示数据组数。

接下来包含 T 组数据，每组数据的格式如下：

第一行包含三个正整数 n, m, k。其中 n, m 表示地图的行数和列数，k 表示机器人执行操作的次数。

第二行包含两个正整数 x_0, y_0 和一个非负整数 d_0。

接下来 n 行，每行包含一个长度为 m 的字符串。保证字符串中只包含 `x` 和 `.` 两个字符。其中，第 x 行的字符串的第 y 个字符代表的位置为 (x,y)。这个位置是 `x` 即代表它是障碍，否则代表它是空地。数据保证机器人初始时所在的位置为空地。

【输出格式】

输出到文件 ***explore.out*** 中。

对于每组数据：输出一行包含一个正整数，表示地图上所有被机器人经过的位置（包括起始位置）的个数。

【样例 1 输入】

```
2
1 5 4
1 1 2
....x
5 5 20
1 1 0
.....
.xxx.
.x.x.
..xx.
x....
```

【样例 1 输出】

```
3
13
```

【样例 1 解释】

该样例包含两组数据。对第一组数据，机器人的状态以如下方式变化：

1. 初始时，机器人位于位置 $(1,1)$，方向朝西（用数字 2 代表）。
2. 第一步，机器人发现它下一步的位置 $(1,0)$ 不在地图内，因此，它会执行“向右转”操作。此时，它的位置仍然为 $(1,1)$，但方向朝北（用数字 3 代表）。
3. 第二步，机器人发现它下一步的位置 $(0,1)$ 不在地图内，因此，它仍然会执行“向右转”操作。此时，它的位置仍然为 $(1,1)$，但方向朝东（用数字 0 代表）。
4. 第三步，机器人发现它下一步的位置 $(1,2)$ 在地图内，且为空地。因此，它会向东走一步。此时，它的位置变为 $(1,2)$，方向仍然朝东。
5. 第四步，机器人发现它下一步的位置 $(1,3)$ 在地图内，且为空地。因此，它会向东走一步。此时，它的位置变为 $(1,3)$，方向仍然朝东。

因此，四步之后，机器人经过的位置有三个，分别为 $(1,1),(1,2),(1,3)$。

对第二组数据，机器人依次执行的操作指令为：向东走到 $(1,2)$，向东走到 $(1,3)$，向东走到 $(1,4)$，向东走到 $(1,5)$，向右转，向南走到 $(2,5)$，向南走到 $(3,5)$，向南走到 $(4,5)$，向南走到 $(5,5)$，向右转，向西走到 $(5,4)$，向西走到 $(5,3)$，向西走到 $(5,2)$，向右转，向北走到 $(4,2)$，向右转，向右转，向南走到 $(5,2)$，向右转，向右转。

【样例 2】

见选手目录下的 ***explore/explore2.in*** 与 ***explore/explore2.ans***。
该样例满足第 $3 \sim 4$ 个测试点的限制条件。

【样例 3】

见选手目录下的 ***explore/explore3.in*** 与 ***explore/explore3.ans***。
该样例满足第 5 个测试点的限制条件。

【样例 4】

见选手目录下的 ***explore/explore4.in*** 与 ***explore/explore4.ans***。
该样例满足第 6 个测试点的限制条件。

【样例 5】

见选手目录下的 ***explore/explore5.in*** 与 ***explore/explore5.ans***。
该样例满足第 $8 \sim 10$ 个测试点的限制条件。

【数据范围】

对于所有测试数据，保证：$1 \le T \le 5, 1 \le n, m \le 10^3, 1 \le k \le 10^6, 1 \le x_0 \le n, 1 \le y_0 \le m, 0 \le d_0 \le 3$，且机器人的起始位置为空地。

学习小组（group）

【题目描述】

班主任计划将班级里的 n 名同学划分为若干个学习小组，每名同学都需要分入某一个学习小组中。观察发现，如果一个学习小组中恰好包含 k 名同学，则该学习小组的讨论积极度为 a_k。

给定讨论积极度 $a_1, a_2, \ldots, a_n$，请你计算将这 n 名同学划分为学习小组的所有可能方案中，讨论积极度之和的最大值。。

【输入格式】

从文件 **group.in** 中读入数据。

第一行，一个正整数 n，表示班级人数。

第二行 n 个非负整数 $a_1, a_2, \ldots, a_n$，表示不同人数学习小组的讨论积极度。

【输出格式】

输出到文件 **group.out** 中。

输出共一行，一个整数，表示所有划分方案中，学习小组讨论积极度之和的最大值。

【样例 1 输入】

```
4
1 5 6 3
```

【样例 1 输出】

```
10
```

【样例 2 输入】

```
8
```

```
2 0 2 5 6 4 3 3 4
```

【样例 2 输出】

```
1 12
```

【数据范围】

对于所有测试点，保证 $1 \leq n \leq 1000$，$0 \leq a_i \leq 10^4$。

团队协作（teamwork）

【题目背景】

在 Farmer John 最喜欢的节日里，他想要给他的朋友们赠送一些礼物。由于他并不擅长包装礼物，他想要获得他的奶牛们的帮助。你可能能够想到，奶牛们本身也不是很擅长包装礼物，而 Farmer John 即将得到这一教训。

Farmer John 的 N 头奶牛（$1 \leq N \leq 10^4$）排成一行，方便起见依次编号为 $1 \ldots N$。奶牛 i 的包装礼物的技能水平为 s_i。她们的技能水平可能参差不齐，所以 FJ 决定把她的奶牛们分成小组。每一组可以包含任意不超过 K 头的连续的奶牛（$1 \leq K \leq 10^3$），并且一头奶牛不能属于多于一个小组。由于奶牛们会互相学习，这一组中每一头奶牛的技能水平会变成这一组中水平最高的奶牛的技能水平。

请帮助 Farmer John 求出，在他合理地安排分组的情况下，可以达到的技能水平之和的最大值。

【输入格式】

从文件 **teamwork.in** 中读入数据。

输入的第一行包含 N 和 K。以下 N 行按照 N 头奶牛的排列顺序依次给出她们的技能水平。技能水平是一个不超过 10^5 的正整数。

【输出格式】

输出到文件 **teamwork.out** 中。

输出 Farmer John 通过将连续的奶牛进行分组可以达到的最大技能水平和。

【样例 1 输入】

```
7 3
1
15
```

```
7
9
2
5
10
```

【样例 1 输出】

```
84
```

【样例 1 说明】

在这个例子中，最优的方案是将前三头奶牛和后三头奶牛分别分为一组，中间的奶牛单独成为一组（注意一组的奶牛数量可以小于 K）。这样能够有效地将 7 头奶牛的技能水平提高至 15、15、15、9、10、10、10，和为 84。