

1、书架

John 最近买了一个书架用来存放奶牛养殖书籍，但书架很快被存满了，只剩最顶层有空余。

John 共有 N 头奶牛($1 \leq N \leq 20,000$)，每头奶牛有自己的高度 H_i ($1 \leq H_i \leq 10,000$)， N 头奶牛的总高度为 S 。书架高度为 B ($1 \leq B \leq S < 2,000,000,007$)。

为了到达书架顶层，奶牛可以踩着其他奶牛的背，像叠罗汉一样，直到他们的总高度不低于书架高度。当然若奶牛越多则危险性越大。为了帮助 John 到达书架顶层，找出使用奶牛数目最少的解决方案吧。

时间限制：10000

内存限制：65536

输入

第 1 行：空格隔开的整数 N 和 B 第 2~ $N+1$ 行：第 $i+1$ 行为整数 H_i

输出

能达到书架高度所使用奶牛的最少数目

样例输入

```
6 40
6
18
11
13
19
11
```

样例输出

```
3
```

2、抓牛

农夫知道一头牛的位置，想要抓住它。农夫和牛都位于数轴上，农夫起始位于点 N ($0 \leq N \leq 100000$)，牛位于点

K ($0 \leq K \leq 100000$)。农夫有两种移动方式：

- 1、从 X 移动到 $X-1$ 或 $X+1$ ，每次移动花费一分钟
- 2、从 X 移动到 $2 \times X$ ，每次移动花费一分钟

假设牛没有意识到农夫的行动，站在原地不动。农夫最少要花多少时间才能抓住牛？

时间限制：2000

内存限制：65536

输入

两个整数，N 和 K

输出

一个整数，农夫抓到牛所要花费的最小分钟数

样例输入

5 17

样例输出

4

3、鸣人和佐助

佐助被大蛇丸诱骗走了，鸣人在多少时间内能追上他呢？

已知一张地图（以二维矩阵的形式表示）以及佐助和鸣人的位置。地图上的每个位置都可以走到，只不过有些位置上有大蛇丸的手下，需要先打败大蛇丸的手下才能到这些位置。鸣人有一定数量的查克拉，每一个单位的查克拉可以打败一个大蛇丸的手下。假设鸣人可以往上下左右四个方向移动，每移动一个距离需要花费 1 个单位时间，打败大蛇丸的手下不需要时间。如果鸣人查克拉消耗完了，则只可以走到没有大蛇丸手下的位置，不可以再移动到有大蛇丸手下的位置。佐助在此期间不移动，大蛇丸的手下也不移动。请问，鸣人要追上佐助最少需要花费多少时间？

时间限制：1000

内存限制：65536

输入

输入的第一行包含三个整数：M，N，T。代表 M 行 N 列的地图和鸣人初始的查克拉数量 T。 $0 < M, N < 200$ ， $0 \leq T < 10$ 后面是 M 行 N 列的地图，其中 @代表鸣人，+代表佐助。*代表通路，#代表大蛇丸的手下。

输出

输出包含一个整数 R，代表鸣人追上佐助最少需要花费的时间。如果鸣人无法追上佐助，则输出-1。

样例输入

样例输入 1

```
4 4 1
#@##
**##
###+
****
```

样例输入 2

```
4 4 2
#@##
**##
###+
****
```

样例输出

样例输出 1

6

样例输出 2

4、献给阿尔吉侬的花束

阿尔吉侬是一只聪明又慵懒的小白鼠，它最擅长的就是走各种各样的迷宫。今天它要挑战一个非常大的迷宫，研究员们为了鼓励阿尔吉侬尽快到达终点，就在终点放了一块阿尔吉侬最喜欢的奶酪。现在研究员们想知道，如果阿尔吉侬足够聪明，它最少需要多少时间就能吃到奶酪。

迷宫用一个 $R \times C$ 的字符矩阵来表示。字符 S 表示阿尔吉侬所在的位置，字符 E 表示奶酪所在的位置，字符 # 表示墙壁，字符 . 表示可以通行。阿尔吉侬在 1 个单位时间内可以从当前的位置走到它上下左右四个方向上的任意一个位置，但不能走出地图边界。

时间限制：100

内存限制：65536

输入

第一行是一个正整数 T ($1 \leq T \leq 10$)，表示一共有 T 组数据。每一组数据的第一行包含了两个用空格分开的正整数 R 和 C ($2 \leq R, C \leq 200$)，表示地图是一个 $R \times C$ 的矩阵。接下来的 R 行描述了地图的具体内容，每一行包含了 C 个字符。字符含义如题目描述中所述。保证有且仅有一个 S 和 E。

输出

对于每一组数据，输出阿尔吉侬吃到奶酪的最少单位时间。若阿尔吉侬无法吃到奶酪，则输出“oop!”（只输出引号里面的内容，不输出引号）。每组数据的输出结果占一行。

样例输入

```
3
3 4
.S..
###.
..E.
3 4
```

```
.S..  
.E..  
....  
3 4  
.S..  
####  
..E.
```

样例输出

```
5  
1  
oop!
```