能否购买

题目ID：623096

限定语言：C++、C++14、Java、C

zzz即将过生日，NQ想要送他一份生日礼物，正巧碰上蛋糕店做活动，NQ义无反顾参与其中。

蛋糕店店长给出一个长度为 n 的整数序列，并另外给出 m 个数，对于这每一个数 b，需要NQ判断是否能在原序列中找出一段长度不超过 k 的连续序列，使其各元素之和大于等于 b。

若对于这 m 个数，NQ都能回答正确，则将获得一折购买大蛋糕的优惠，但是NQ不会做这题，所以向致力于 ACM 的你求助，希望能够一起给zzz过生日。

输入描述:

第一行输入三个用空格隔开的整数 `n, m, k`，具体含义如题所述。
第二行一共 `n` 个整数 a_i，表示这个序列中第 `i` 个元素的值。
接下去 `m` 行，每行一个整数 `b`，表示询问。
$1 \le n, m, k \le 10^6$, $|a_i| \le 10^3$, $|b| \le 10^9$

输出描述:

对于每一个整数 `b`，如果原序列 `a` 中存在满足题意要求的序列，则输出 `"YES"`，否则输出 `"NO"`.（均不包含引号）

示例1

输入

```
5 3 2
1 -1 2 3 -2
1
5
6
```

输出

```
YES
YES
NO
```

吃面包

题目ID：623094

限定语言：C++、C++14、Java、C

在校赛期间各位 ACM 大佬的帮助下，zzz终于成功解决了买面包的问题，获得了 SZ 市 IT 牌面包店的至尊权限。想吃什么面包，就买什么面包。

现在，zzz迷上了一款奶油面包，不是说面包有多好吃，而是奶油的吸引力太强。而这款奶油面包也有着自己的独特之处：它的奶油分布并不是均匀的，而是随机在面包内某处分布有部分奶油。

由于zzz刚跟他的队友吃完羊排，吃不下面包，但是又抵挡不住奶油的诱惑，所以想要用吸管吸奶油，为了便于不浪费太多时间，zzz又向Enal大魔法师借了一副魔法眼镜，获得到透视的功能，能够清晰地看见面包内部某个位置存在多少奶油，假定1克奶油能够提供给zzz 1点饱腹感，那么，在zzz不会撑死的前提下，求能够使zzz获得最大饱腹感的不同奶油吸取方案数，由于数量可能很大，所以请将答案对 $10^9 + 7$ 取模。

输入描述:

> 第一行输入两个整数用空格隔开 `n, m,` 表示zzz通过透视眼镜发现面包内一共有 `n` 处存在奶油（这 `n` 出奶油各自独立，两两不相连），zzz再获得超过 `m` 点饱腹感就有撑死的风险。
>
> 接来下 `n` 行，每行一个整数 `g`，表示某个位置存在 `g` 克奶油（如果zzz一旦选择某个位置开始吸奶油，则一定会吸完）。
>
> $0 < n, m, g \leq 1000$

输出描述:

> 输出一个整数，表示方案数对 $10^9 + 7$ 取模的结果。

示例1

输入

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

输出

```
2
```

说明

> zzz可以吃第二个位置和第三个位置的奶油获得5点饱腹感，也可以只吃第四个位置获得5点饱腹感，所以一共有两种方案。

多组输入输出斐波那契

题目ID：623093

限定语言：C++、C++14、Java、C

求斐波那契数列第 n 项对 998244353 取模的结果。

所谓斐波那契数列指的是数列：1，1，2，3，5，8，13，21，……。即数列满足递推公式：

$F_1 = F_2 = 1$，$F_n = F_{n-2} + F_{n-1}$ $(n \geq 3, n \in N^*)$

输入描述:

第一行输入一个整数 T $(1 \leq T \leq 10^6)$

接下去 T 行，每行一个整数 n $(1 \leq n \leq 10^6)$

输出描述:

对于每个整数 n，输出斐波那契数列第 n 项对 998244353 取模的结果。

示例1

输入

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

输出

```
1
1
2
3
5
```

题目ID：623092
限定语言：C++、C++14、Java、C

在刚刚结束的第44届ACM-ICPC国际大学生程序设计竞赛亚洲区域赛（银川站）中，我校成功获得了铜牌，这也是校史首次在该赛事中获奖。

然而，比赛结束后，队内的两位主力选手 lzh、zcy却不太开心，因为我校排在122名，位居铜牌区首位，只差一名就可以拿到银牌。

ACM-ICPC区域赛金银铜奖的规则大致如下：
金银铜奖分别为有效参赛队总数的10%、20%、30%，即排名前10%为金奖，11%-30%为银奖，31%-60%铜奖。有些巨佬参赛队会打星参赛，即参与排名，但颁奖时不计入在内。
现在给出有效参赛队总数和打星队伍的排名，请你计算一下金银铜奖实际依次对应的名次

输入描述:

> 第一行两个整数，用空格隔开，表示有效参赛队总数 `N` 和打星队伍的数量 `K`
> 第二行有 `K` 个整数，用空格隔开，依次表示 `K` 支打星队伍的排名（注意：这 `K` 支队伍排名无序）
>
> $120 \leq N \leq 500$，$0 \leq K \leq 200$ ，为方便计算，保证 `N` 是 `10` 的整数倍

输出描述:

> 输出由三行构成，每行两个整数，中间由空格隔开
> 三行依次对应金奖、银奖、铜奖。每行的两个整数依次代表第一个获得金/银/铜奖的队伍排名，和最后一个获得金/银/铜奖的队伍排名

示例1

输入

```
400 1
88
```

输出

```
1 40
41 121
122 241
```

说明

> 样例表示有400支有效参赛队，可算出应有40个金奖、80个银奖、120个铜奖。1支打星队，排名为第88名
> 则排名1-40名为金奖
> 排名41-120的队伍中有一支打星队，所以银奖跳过该打星队顺延一名，排名41-121的队伍获得银奖
> 铜奖依次顺延，排名122-241的队伍获得铜奖

大佬数

题目ID：623090

限定语言：C++、C++14、Java、C

小咸鱼最近对质数非常感兴趣，他自己定义了一个概念————大佬数

首先算出某个数 N 的质因子，再对这些质因子求和，若和大于610，则认为是大佬数

输入描述:

> 输入只有一行，一个整数 `N`
> $0 \leq N \leq 10000$

输出描述:

> 输出一行字符串，如果 `N` 是大佬数，则输出`"a sdl wsl"`，否则输出`"tcl"`。（只用输出引号内的内容）

示例1

输入

```
8872
```

输出

```
a sdl wsl
```

装水容器

题目ID：623089

限定语言：C++、C++14、Java、C

给定 n $(2 \leq n \leq 100)$ 个非负整数 $a_1, a_2, ..., a_n$ ，每个数代表第 i 个木板的高度 h $(1 \leq h \leq 10000$ 单位 m），忽略木板体积，木板面积为 1m。现在把这些木板间隔 1m 均匀的放在一个大小合适的容器内。你需要找出两条木板，使得他们之间构成的容器可以容纳最多的水。

俩木板选定之后会拆除其他木板，你只用计算这俩木板组成容器的容积。

输入描述:

> 题目有多组数据，第一行 `T` 表示数据组数
> 每组数据第一行 `n`
> 接下来一行为 `n` 个木板的高度，空格分隔。

输出描述:

> 给出每组能容纳最多的水的体积。

示例1

输入

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

输出

```
49
```

嵌套深度

题目ID：623088

限定语言：C++、C++14、Java、C

在一段C语言代码中，括号是可以嵌套的。例如：

1. “()”表示没有嵌套；
2. “(())”表示嵌套1层；
3. “((((()))))”嵌套了3层；
4. “(((())(((())))()” 嵌套最深为3层；
5. “)()())”的括号没有闭合。

你的任务就是输出这个括号字符串里面括号嵌套的最深深度，如果括号没有结合，就输出 -1 。

输入描述:

> 题目有多组数据，第一行 $n(1 \leq n \leq 100)$ 表示数据组数
> 接下来 `n` 行为括号组成的表达式 `s` 长度小于`100`，中间不含空格单纯由括号组成。

输出描述:

> 每个表达式括号的嵌套最深深度，如果括号最后没有闭合，输出`-1`

示例1

输入

```
2
()(())()
)(
```

输出

```
1
-1
```

细胞游戏

题目ID：622989

限定语言：C++、C++14、Java、C

给定一个 m × n 个格子的二维面板，每一个格子都可以看成是一个细胞。每个细胞具有一个初始状态， 1 即为活细胞, 0 为死细胞。每个细胞与其八个相邻位置（水平，垂直，对角线）的细胞都遵循以下四条生存定律：

1.如果活细胞周围八个位置的活细胞数少于两个，则该位置活细胞死亡；
2.如果活细胞周围八个位置有两个或三个活细胞，则该位置活细胞仍然存活；
3.如果活细胞周围八个位置有超过三个活细胞，则该位置活细胞死亡；
4.如果死细胞周围正好有三个活细胞，则该位置死细胞复活；

根据当前状态，写一个程序来计算面板上细胞的下 q 个（q 次更新后的）状态。下一个状态是通过将上述规则同时应用于当前状态下的每个细胞所形成的，其中细胞的出生和死亡是同时发生的。

面板上所有格子需要同时被更新，你不能先更新某些格子，然后使用它们的更新后的值再更新其他格子。

$3 \leq m, n \leq 100$
$1 \leq q \leq 10$

输入描述:

> 第一行给出 `m,n,q` 空格分隔
> 接下来给出 `m` 行，每行 `n` 个的二维面板

输出描述:

> 输出 `q` 次更新后面板上所有细胞的状态

示例1

输入

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

输出

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