

Problem A. 90-degree Rotations

Input file: `rotate.in`
Output file: `rotate.out`
Time limit: 2 seconds
Memory limit: 512 mebibytes

给定一张 $H \times W$ 的网格图，行标号为 1 到 H ，列标号为 1 到 W 。

网格图的某些格子中存在金币。如果 $S_{i,j} = 'o'$ ，那么 (i,j) 就有一个金币，否则 (i,j) 没有金币。

小 C 会在某一个**有金币的格子**中放置一个机器人，并选择一个初始方向（上/下/左/右）。然后，机器人将会不断重复如下操作：

- 收集机器人所在格子的金币。
- 向左或向右旋转 90° 。
- 向前走任意正整数格，到达另一个**有金币的格子**。
- 如果无法走到另一个有金币的格子，机器人就会停止执行操作。

请你确定小 C 的机器人是否能收集所有的金币。

Constraints

- $T \leq 10$
- $2 \leq H \leq 100$
- $2 \leq W \leq 100$
- $S_{ij} = 'o'$ 或 $'x'$
- 存在合法的 i,j 使得 $S_{i,j} = 'o'$

Input

输入共 T 组。

每组输入共 $H + 1$ 行。

第一行包含两个整数 H, W 。

接下来的 H 行包含字符矩阵 $S_{i,j}$ 。

Output

输出共 T 行。

输出 `'Possible'`，如果存在一种方式能收集所有金币。否则输出 `'Impossible'`。

Examples

rotate.in	rotate.out
2 2 9 OXXXXXXO OXOXXXXO 5 5 XOXX XOXO XXOX OOXO XOXXX	Possible Possible
2 6 7 XOOOXOO XOOOXOO OXOXXOO OXOXOXX XXOXXOX OXXOXOO 6 6 OXXXXO XOOOXX XOOOXX XOOOXX XXOXX OXXXXO	Impossible Impossible

Subtask 1 (40 points)

金币数量不超过 8。

Subtask 2 (20 points)

金币数量不超过 16。

Subtask 3 (20 points)

$H = 2$

Subtask 4 (20 points)

没有额外的约束。

Note

本题采用子任务形式打包测评，测评时开启 -O2 优化。

Problem B. Multisets

Input file: `multiset.in`
Output file: `multiset.out`
Time limit: 2 seconds
Memory limit: 512 mebibytes

我们说一个可重集 \mathcal{A} 比可重集 \mathcal{B} 小，当且仅当对于两个可重集中出现次数不同的最小元素 x ，元素 x 在 \mathcal{A} 中出现次数更多。

例如，可重集 $\{1, 2, 3\}$ 就比可重集 $\{1, 3, 3, 5\}$ 小，类似的， $\{1, 1, 4, 4\}$ 小于 $\{1, 1, 4\}$ 。

小 C 给你一个长度为 n 的正整数序列 S 。考虑 S 的所有连续子序列，可以把它们分别看做一个可重集。（也就是说，恰好存在着 $\frac{n(n+1)}{2}$ 个可重集）

小 C 想知道第 k 小可重集，想请你帮她找到答案。

Constraints

- $1 \leq S_i \leq n \leq 100000$
- $1 \leq k \leq \frac{n(n+1)}{2}$

Input

输入共两行。

第一行包含两个整数 n, k 。

接下来的一行包含序列 S_i 。

Output

输出共一行。

输出你找到的可重集，**请输出可重集排序后的结果**。

Examples

multiset.in	multiset.out
6 5 1 2 1 3 5 1	1 1 2
8 9 6 3 4 4 1 4 5 8	1 4 4 4 5 8
50 233 21 7 15 17 43 38 32 12 41 6 31 27 16 25 41 32 22 11 2 30 26 1 9 43 39 18 1 30 4 2 29 40 48 42 1 11 7 45 25 43 14 17 8 16 36 9 29 18 34 17	1 1 1 2 2 4 7 9 11 11 14 16 17 18 22 25 25 26 27 29 30 30 31 32 39 40 41 42 43 43 45 48

Subtask 1 (40 points)

$$n \leq 100$$

Subtask 2 (20 points)

$$n \leq 1000$$

Subtask 3 (10 points)

$$n \leq 30000, k \leq 100$$

Subtask 4 (10 points)

$$n \leq 30000, S_i \leq 20$$

Subtask 5 (15 points)

$$n \leq 30000$$

Subtask 6 (5 points)

没有额外的约束。

Note

提示：保证数据随机生成，但这对你的解题并不是必需的。

本题采用子任务形式打包测评，测评时开启 -O2 优化。

Problem C. Fibonacci's Nightmare

Input file: `nightmare.in`
 Output file: `nightmare.out`
 Time limit: 2 seconds
 Memory limit: 512 mebibytes

定义一个随机线性递推序列（即 *random linear recursive sequence, RLRS*）是一个由如下方式生成的非负整数序列。一开始，令 $a_0 = 1$ 。然后对于从 1 开始的每一个 n ，独立且等概率在 $[0, n-1]$ 中随机 k 个非负整数 i_1, i_2, \dots, i_k ，令 $a_n = \sum_{j=1}^k a_{i_j}$ 。

接下来是一个例子。令 $k = 2$ ，那么有 $a_1 = a_0 + a_0 = 2$ ，之后， a_2 的值等概率从 $a_0 + a_0, a_0 + a_1, a_1 + a_0, a_1 + a_1$ 中选择。

小 C 解决了 $k = 2$ 时 a_n^2 的数学期望。她想问你对于更大的 k ， $\mathbb{E}(a_n^k)$ 在模 998244353 下的值。

Constraints

- $2 \leq n \leq 100000$
- $2 \leq k \leq 10$

Input

输入共一行，包含两个整数 n, k 。

Output

输出共一行，答案在模 998244353 下的值。

Examples

<code>nightmare.in</code>	<code>nightmare.out</code>
2 2	499122186
4 2	214899855
147 2	976561559
147 5	93471701
2019 10	950838005

Subtask 1 (30 points)

$$n \leq 5, k = 2$$

Subtask 2 (10 points)

$$n \leq 10, k = 2$$

Subtask 3 (20 points)

$$n \leq 200, k = 2$$

Subtask 4 (10 points)

$$n \leq 1000, k = 2$$

Subtask 5 (10 points)

$$k = 2$$

Subtask 6 (15 points)

$$k \leq 5$$

Subtask 7 (5 points)

没有额外的约束。

Note

本题采用子任务形式打包测评，测评时开启 `-O2` 优化。