1 复制程序

题目描述

一个程序最初只在一台机器上，现需要复制到其他机器上，总机器数为 N（其中一台机器已有该程序）。
只能通过直连电缆点对点复制，电缆数量为 K，设复制一次耗时 1 小时，求出最小复制时间。

输入格式

输入仅一行，包含 2 个整数 N，K，分别表示总机器数以及电缆数量。
数据范围：
$1 \le N \le 1,000,000,000$
$1 \le K \le 1,000,000,000$

输出格式

输入一行，包含一个整数，表示最小耗时（单位为小时）。

示例 1

输入

```
8 3
```

输出

```
4
```

2 手动执行

题目描述

多多鸡在公司负责一个分布式任务执行平台的维护。夏天到了，平台的服务器都因中暑而无法执行任务了。所以多多鸡必须自己手动来执行每个任务！
现在一共有 N 个待执行的任务，每个任务多多鸡需要 Pi 的时间完成执行。同时，任务之间可能会有一些依赖关系。比如任务 1 可能依赖任务 2 和任务 3，那么任务 1 必须在任务 2 和任务 3 都执行完成后才能执行。
多多鸡只有一个人，所以同时只能执行一个任务，并且在任务完成之前不能暂停切换去执行其他任务。为了提升平台用户的使用体验，多多鸡希望最小化任务的平均返回时长。一个任务的返回时长定义为任务执行完成时刻减去平台接收到该任务的时刻。在零时间，所有 N 个任务都已经被平台接收。
请你帮多多鸡安排一下任务执行顺序，使得平均返回时长最小。

输入格式

第一行包含 2 个正整数 N 、M ，分别表示任务数量以及 M 个任务依赖关系。
第二行包含 N 个正整数，第 i (1 <= i <= N) 个数表示第 i 个任务的处理时间 Pi 。
接下来的 M 行，每行表示一个任务依赖关系。每行包含 2 个整数 Ai (1 <= Ai <= N)、Bi (1 <= Bi <= N)，表示第 Bi 个任务依赖于第 Ai 个任务。数据保证不会出现循环依赖的情况。
数据范围：
1 <= N <= 1000
1 <= M <= 50000
1 <= Pi <= 10000

输出格式

输出一行，包含 N 个整数(两两之间用一个空格符分隔)，其中第 i (1 <= i <= N) 个整数表示多多鸡执行的第 i 个任务的编号。若有多种可行方案，则输出字典序最小(优先执行编号较小的任务)的方案。

示例 1

输入

```
5 6
1 2 1 1 1
1 2
1 3
1 4
2 5
3 5
4 5
```

输出

```
1 3 4 2 5
```

3 AB 字符串

题目描述

给你两个正整数 N 和 K，希望找到满足下面条件的字符串。

- 字符串的长度为 N，并且仅由’A’和’B’两个字符组成
- 该字符串 s 刚好有 K 对（i，j），（0 <= i < j <= N-1）满足 s[i] = 'A'且 s[j] = 'B’

如果这样的字符串存在，找到并返回字典序最大的一个，否则返回字符串 “-”。

输入格式

第一行是一个数字 N
第二行是一个数字 K
2 <= N <= 50, 0 <= K <= N(N-1)/2

输出格式

输出满足要求的字典序最大的字符串，否则输出字符串”-“(不需要带双引号)

示例 1

输入

```
7
3
```

输出

```
BBBABBB
```

示例 2

输入

```
11
45
```

输出

```
-
```

4 非凡字符串

题目描述

对于一个字符串 S，我们定义它的一个长度为 3 的子串 $\{S_i, S_{i+1}, S_{i+2}\}$ $(0 <= i < |S| - 2)$ 为***非凡子串***当 $S_i = S_{i+1} = S_{i+2}$ 。比如 S = "aaabbbb" 有三个非平凡子串："aaa" (i = 0), "bbb" (i = 3), "bbb" (i = 4)。我们定义一个字符串 S 为***非凡字符串***当 S 有且只有一个***非凡子串***。比如字符串 "aaabb" 是一个非凡字符串，但 "aaaa", "aaabbb", "abc" 都不是非凡字符串。

如果只使用小写英文字母的前 S 个字符，那么存在多少个长度为 L 的非凡字符串，给出结果模 10^9+7 的值

输入格式

输入一行，包含 2 个整数 L 和 S，分别表示字符串长度 L 以及字符集大小 S 。

数据范围：

1 <= L, S <= 26

输出格式

输出一行，表示模 10^9+7 的值

示例 1

输入

```
3 7
```

输出

```
7
```