

玩具火车 (train)

输入文件名	train.in
输出文件名	train.out
单测试点时限	1.0 秒
内存限制	512 MB
测试点数目 / 单测试点分值	10/10
提交源程序文件名	
对于 C 语言	train.c
对于 C++ 语言	train.cpp
对于 Pascal 语言	train.pas

● 题目描述

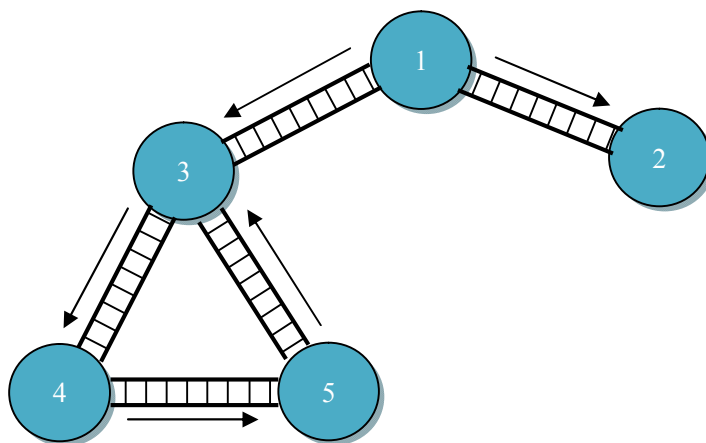
明明得到了一辆新的玩具火车。

这辆玩具火车行驶的地图上有许多的站台和连接这些站台的有向轨道。火车只能从站台出发，沿着有向轨道行驶，到达另外的站台。

开始时，明明可以任意指定一个站台作为火车的出发站台。火车每到达一个站台（包括出发站台），都有可能存在超过 1 条的轨道可以行驶。在这个时候，明明可以从这些轨道中任选一条作为火车接下来的行驶轨道。如果没有道路可选择，那么火车就将停止行驶。站台与轨道没有经过次数的限制。

好奇的明明发现：有的时候，明明能够选择合适的路线使得火车一直不停地跑下去。他对此很感兴趣，因此想请你告诉他，火车究竟从哪些站点出发能够有机会不停地行驶。

一个具体的例子如下：火车行驶的地图上共有 5 个站点和 5 条有向轨道，如果我们将站台从 1~5 编号，用 (a, b) 表示一条从站台 a 到站台 b 的有向轨道，那么这 5 条轨道分别为 $(1, 2)$, $(1, 3)$, $(3, 4)$, $(4, 5)$, $(5, 3)$ 。地图如下图所示：



如果火车：

- I. 从站台 1 出发，使火车沿路线 $1 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 3 \rightarrow 4 \rightarrow 5 \cdots$ 行驶，火车能不停地跑下去。
- II. 从站台 2 出发，没有道路可供明明选择，因此火车无法行驶。
- III. 从站台 3 出发，使火车沿路线 $3 \rightarrow 4 \rightarrow 5 \rightarrow 3 \rightarrow 4 \rightarrow 5 \cdots$ 行驶，火车能不停地跑下去。
- IV. 从站台 4 出发，使火车沿路线 $4 \rightarrow 5 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 3 \cdots$ 行驶，火车能不停地跑下去。
- V. 从站台 5 出发，使火车沿路线 $5 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 3 \rightarrow 4 \cdots$ 行驶，火车能不停地跑下去。

因此，火车从站台 1, 3, 4, 5 出发都有机会能不停地行驶。

● 输入格式

输入文件第一行包含两个正整数 n 和 m ，分别表示站台的数量与有向轨道的数量。为了方便，我们将站台从 1 至 n 编号。

接下来 m 行，每行包含 2 个正整数 a, b ，表示有一条连接站台 a, b 的有向轨道，方向为从站台 a 至站台 b 。

● 输出格式

输出文件包含 n 行，每行仅包含 1 个整数，且该整数要么为 0，要么为 1。若火车从站台 i 出发，最终有机会能无限地行驶下去，那么第 i 行的整数为 1，否则为 0。

● 样例输入与输出

样例 1 输入：

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

样例 1 输出：

```
1
0
1
1
1
```

样例 1：见用户目录下 `train1.in` 与 `train1.ans`。

样例 2：见用户目录下 `train2.in` 与 `train2.ans`。

样例 3：见用户目录下 `train3.in` 与 `train3.ans`。

● 数据范围与约定

所有测试点统一拥有如下限制：

- 对于任意站台 a, b ，保证不存在超过 1 条从站台 a 至站台 b 的有向轨道。
- 可能会存在从站台 a 出发，回到站台 a 的有向轨道。
- 整个地图可能不连通。

除了以上已给出的限制外，单个测试点的具体数据范围及特殊说明见下表：

Problem 1.Train

测试点编号	$n=$	$m=$	特殊说明
1	8	20	无
2	300	500	
3	300	500	
4	2×10^3	2×10^3	保证地图连通，且地图仅包含 1 个简单环
5	2×10^3	5×10^3	无
6	2×10^3	5×10^3	
7	5×10^4	5×10^4	保证地图连通，且地图仅包含 1 个简单环
8	2×10^5	5×10^5	无
9	5×10^5	10^6	
10	10^6	10^6	