Question **2**

Correct

Mark 1.00 out of 1.00

Trong một giải đấu bóng đá gồm n đội bóng, đánh số từ 1 đến n. Mỗi trận đấu có hai đội thi đấu với nhau cho đến khi phân biệt thắng thua (ví dụ: hiệp phụ, đá luân lưu). Sau khi giải đấu kết thúc, ban tổ chức muốn xếp hạng các đội theo quy tắc sau:

- Hạng được tính từ 1, 2, 3, ...
- Đội không thua trận nào xếp hạng 1
- Nếu đội A đá **thắng** đội B thì hạng của đội A **nhỏ hơn** hạng của đội B.
- Nếu một đội có thể nhận nhiều hạng khác nhau thì chọn hạng nhỏ nhất.

Hoặc bạn cũng có thể sử dụng định nghĩa sau:

$$\textbf{Hạng(v) = max \{Hạng(u) \} + 1}$$

với u là đội thắng v.

Hãy giúp ban tổ chức viết chương trình xếp hạng cho các đội. Giả sử không xảy ra trường hợp A thắng B, B thắng C ,..., Z thắng A.

Đầu vào (Input):

Dữ liệu đầu vào được nhập từ bàn phím với định dạng:

- Dòng đầu tiên chứa 2 số nguyên **n** và **m**, tương ứng là số đội và số trận đấu.
- **m** dòng tiếp theo mỗi dòng chứa 2 số nguyên **u v** mô tả kết quả trận đấu: **u** thắng, **v** thua.

Đầu ra (Output):

- In ra màn hình hạng của các đội bóng theo số thứ tự của đội trên cùng 1 dòng, mỗi đội cách nhau 1 khoảng trắng.

<Hạng đội 1> <Hạng đội 2> ... <Hạng đội n>

Xem thêm ví dụ bên dưới. Trong ví dụ đầu tiên ta có: hạng của 1 = 1, hạng của 2 = 3 và hạng của 3 = 2.

Trong ví dụ 2: đội 2 có thể nhận hạng 2 hoặc 3 nên xếp hạng 2, tương tự như thế cho đội 4.

For example:

Input	Result
3 2 1 3 3 2	1 3 2
7 10 1 2 1 3 1 4 2 3 2 6 3 7 4 5 5 3 5 7 6 7	1 2 4 2 3 3 5

Answer: (penalty regime: 10, 20, ... %)

```
#include <stdio.h>
#define MAX_N 100
typedef int ElementType;
int r[MAX_N];

//---------------------Queue-----------------------
// typedef struct{
//  ElementType data[MAX_N];
//  int front, rear;
// }Queue;

// void init_queue (Queue *pQ){
```

```
13 //  pQ->front = 0;
14 //  pQ->rear = -1;
15 // }
16
17 // int front (Queue *pQ){
18 //  return pQ->data[pQ->front];
19 // }
20
21 // void enqueue (Queue *pQ, ElementType x){
22
```

Precheck Check

	Input	Expected	Got	
✔	3 2 1 3 3 2	1 3 2	1 3 2	✔
✔	7 10 1 2 1 3 1 4 2 3 2 6 3 7 4 5 5 3 5 7 6 7	1 2 4 2 3 3 5	1 2 4 2 3 3 5	✔
✔	7 12 1 2 1 3 2 4 2 5 2 6 3 2 3 5 3 6 4 7 5 7 6 4 6 5	1 3 2 5 5 4 6	1 3 2 5 5 4 6	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

◄ 003. Xếp hạng các đỉnh

Jump to...

005. Quản lý dự án ►