

# Mau Pulang

## Deskripsi

Hachikuji Mayoi adalah seorang siswi yang sangat sering tersesat. Dia selalu saja buta arah dan tidak bisa kembali ke rumah. Ditambah, terkadang Mayoi hanya dapat melewati jalan tertentu yang bersifat satu arah saja.

Maka dari itu, kamu memutuskan untuk membantu Mayoi untuk menentukan semua node yang memungkinkan Mayoi untuk kembali ke node tersebut, yang mana membantu Mayoi untuk menentukan tempat tinggalnya.

## Format Masukan

Baris pertama berisikan tiga buah bilangan  $N, M$  ( $1 \leq N \leq 100$ ,  $0 \leq M \leq N \times (N - 1)$ ) yang masing-masing menyatakan jumlah node dan jumlah edge.

$M$  baris berikutnya untuk masing-masing barisnya berisikan 2 buah bilangan bulat  $A_i B_i$  ( $1 \leq A_i, B_i \leq N$ ,  $A_i \neq B_i$ ) yang menyatakan bahwa jika Mayoi berada pada posisi  $A_i$ , maka ia bisa pergi ke posisi  $B_i$ .

## Format Keluaran

Keluarkan satu baris yang berisikan semua node yang memungkinkan Mayoi untuk pulang kembali ke node tersebut.

Jika tidak ada node yang memungkinkan Mayoi untuk kembali ke node tersebut, maka cukup keluarkan -1.

### Contoh Masukan 1

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

### Contoh Keluaran 1

```
1 3 4 6
```

### Contoh Masukan 2

```
3 2
1 2
2 3
```

### Contoh Keluaran 2

```
-1
```

### Contoh Masukan 3

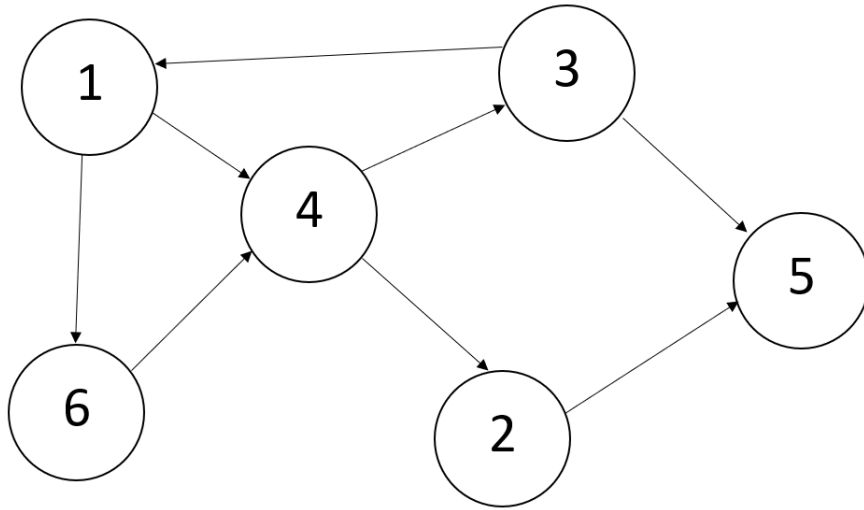
```
4 4
1 2
2 1
3 4
4 3
```

### Contoh Keluaran 3

```
1 2 3 4
```

## Penjelasan

Graph pada contoh pertama dapat dilihat pada gambar berikut



Dapat diperhatikan bahwa terdapat 4 node yang dapat membawa Mayoi kembali ke node tersebut:

- Node 1:  $1 \rightarrow 4 \rightarrow 3 \rightarrow 1$
- Node 3:  $3 \rightarrow 1 \rightarrow 4 \rightarrow 3$
- Node 4:  $4 \rightarrow 3 \rightarrow 1 \rightarrow 4$
- Node 6:  $6 \rightarrow 4 \rightarrow 3 \rightarrow 1 \rightarrow 6$ .

Untuk node 2, Mayoi hanya dapat menuju node 5 yang mana node 5 tidak memiliki out-degree, sehingga Mayoi tidak dapat kembali.

## Petunjuk

Lakukan DFS sebanyak  $N$  kali, dari node 1 sampai  $N$ . Untuk setiap DFS, cek apakah node mengunjungi node awal atau tidak. Jika stack sudah kosong tanpa mengunjungi node tersebut kembali, maka node tersebut bukanlah solusi.