スーパーツリーをつなげ (supertrees)

ガーデンズ・バイ・ザ・ベイはシンガポールにある国立公園である. 公園には n 本のタワーがあり, スーパーツリーと呼ばれている. これらのタワーには 0 から $n-1$ までの番号が付けられている. 我々は今, 0 **本以上の** 渡り廊下を建設したい. それぞれの渡り廊下は 2 つの異なるタワーをつないでいて, **双方向に** 移動することができる. また, どの 2 つのタワーのペアの間にも, 2 本以上の渡り廊下は存在しない.

タワー x からタワー y へのパスとは, 次の条件を満たすような長さ 1 以上のタワーの列のことである.

- 列の最初の要素は x である.
- 列の最後の要素は y である.
- 列のすべての要素は **互いに異なる** .
- 列のすべての隣り合った要素(タワー)は, 渡り廊下によって結ばれている.

注意として, 定義より, あるタワーからそのタワー自身へのパスはただ 1 つである. また, タワー x からタワー y への異なるパスの数は, タワー y からタワー x への異なるパスの数と等しい.

デザイン担当の建築家は渡り廊下のデザインに関して, すべての $0 \leq i, j \leq n-1$ について, タワー i からタワー j への異なるパスの数がちょうど $p[i][j]$ になるようにしたいと言っている. ただし, $0 \leq p[i][j] \leq 3$ を満たす.

建築家の要求を満たすような渡り廊下の組み合わせを構築せよ. もしくは, 要求を満たすような組み合わせがないことを知らせよ.

実装の詳細

あなたは以下のプロシージャを実装しなければならない.

```
int construct(int[][] p)
```

- p: $n \times n$ の二次元配列であり, 建築家の要求を表している.
- 構築することができるのならば, ちょうど 1 回 `build` プロシージャ(後述)を呼び出して渡り廊下の組み合わせを報告し, その後 1 を返さなければならない.
- 構築するのができないのならば, `build` プロシージャを呼び出すことなく 0 を返さなければならない.
- このプロシージャはちょうど 1 回呼び出される.

`build` プロシージャは以下のように定義されている.

```
void build(int[][] b)
```

- b: $n \times n$ の二次元配列であり，タワー i とタワー j の間に渡り廊下があるならば $b[i][j] = 1$ となり，そうでないならば $b[i][j] = 0$ となる．
- 注意として，すべての $0 \leq i, j \leq n-1$ で $b[i][j] = b[j][i]$ を満たし，かつ，すべての $0 \leq i \leq n-1$ で $b[i][i] = 0$ を満たす必要がある．

入出力例

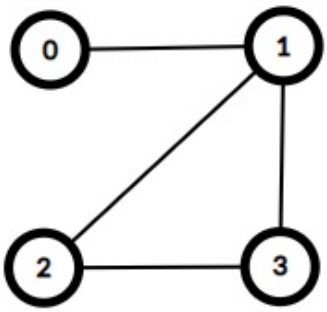
入出力例1

以下のような呼び出しを考える．

```
construct([[1, 1, 2, 2], [1, 1, 2, 2], [2, 2, 1, 2], [2, 2, 2, 1]])
```

この呼び出しは，タワー 0 からタワー 1 の間にはちょうど 1 つのパスが存在し，それ以外の $0 \leq x < y \leq 3$ を満たすようなタワー x とタワー y の間にはそれぞれちょうど 2 つのパスが存在するべきであることを意味する．

この条件を満たすには，4 つのタワーの組 $(0, 1)$, $(1, 2)$, $(1, 3)$, $(2, 3)$ をそれぞれ渡り廊下で結べばよい．



この答えを報告するために，`construct` プロシージャは以下の呼び出しをしなければならない．

- `build([[0, 1, 0, 0], [1, 0, 1, 1], [0, 1, 0, 1], [0, 1, 1, 0]])`

そして 1 を返さなければならない．

この入力では，この要求を満たす複数の渡り廊下の組み合わせが存在する．このうちどれを回答したとしても正解となる．

入出力例2

以下のような呼び出しを考える．

```
construct([[1, 0], [0, 1]])
```

この呼び出しは，2 つのタワーの間には移動する手段がないことを意味する．これは，1 本も渡り廊下がないことによってのみ満たされる．

そのため，`construct` プロシージャは以下の呼び出しをしなければならない．

- `build([[0, 0], [0, 0]])`

その後，`construct` プロシージャは 1 を返さなければならない．

入出力例3

以下のような呼び出しを考える．

```
construct([[1, 3], [3, 1]])
```

この呼び出しは，タワー 0 からタワー 1 へのパスはちょうど 3 つであるべきことを意味する．この要求は満たすことができない．そのため，`construct` プロシージャは `build` プロシージャの呼び出しを行うことなく，0 を返さなければならない．

制約

- $1 \leq n \leq 1000$
- $p[i][i] = 1$ $(0 \leq i \leq n-1)$
- $p[i][j] = p[j][i]$ $(0 \leq i, j \leq n-1)$
- $0 \leq p[i][j] \leq 3$ $(0 \leq i, j \leq n-1)$

小課題

1. (11 点) $p[i][j] = 1$ $(0 \leq i, j \leq n-1)$
2. (10 点) $0 \leq p[i][j] \leq 1$ $(0 \leq i, j \leq n-1)$
3. (19 点) $p[i][j] = 0$ もしくは $p[i][j] = 2$ $(0 \leq i, j \leq n-1, i \neq j)$
4. (35 点) $0 \leq p[i][j] \leq 2$ $(0 \leq i, j \leq n-1)$ かつ，最低でも 1 つは要求を満たすような渡り廊下の組み合わせが存在する．
5. (21 点) $0 \leq p[i][j] \leq 2$ $(0 \leq i, j \leq n-1)$
6. (4 点) 追加の制約はない．

採点プログラムのサンプル

採点プログラムのサンプルの入力形式は以下のとおりである．

- 1 行目: n
- $2+i$ 行目 $(0 \leq i \leq n-1)$: $p[i][0]$ $p[i][1]$ $\ldots$ $p[i][n-1]$

採点プログラムのサンプルの出力形式は以下のとおりである．

- 1 行目: `construct` プロシージャの戻り値

もし `construct` プロシージャの戻り値が 1 である場合，採点プログラムのサンプルは追加で以下を表示

する.

- $2+i$ 行目 ($0 \leq i \leq n-1$): $b[i][0]$ $b[i][1]$ $\ldots$ $b[i][n-1]$