スフィンクスの謎 (Sphinx's Riddle)

ギザの大スフィンクスはあなたに謎解きを用意した．あなたには N 頂点からなるグラフが与えられる．頂点には 0 から $N-1$ までの番号が付けられている．グラフには M 本の辺があり，0 から $M-1$ までの番号が付けられている．それぞれの辺は相異なる 2 つの頂点を双方向に結ぶ．具体的には，与えられるグラフにおいて，辺 j $(0 \leq j < M)$ は頂点 $X[j]$ と $Y[j]$ を結ぶ．どの相異なる 2 つの頂点についても，それらを結ぶ辺は高々 1 本しか存在しない．2 つの頂点が辺で結ばれているとき，それらの頂点は **隣接している** という．

頂点の列 $v_0, v_1, \ldots, v_k$ $(k \geq 0)$ は，$0 \leq l < k$ を満たすすべての l に対し頂点 v_l, v_{l+1} が隣接しているとき，**パス** と呼ばれる．パス $v_0, v_1, \ldots, v_k$ は頂点 v_0 と v_k を **繋いでいる** という．あなたに与えられるグラフにおいては，どの相異なる 2 つの頂点についても，それらを繋ぐパスが存在する．

0 から N までの番号が付けられた $N+1$ 種類の色がある．色 N は特別であり，**スフィンクスの色** と呼ばれる．それぞれの頂点にはある 1 つの色が塗られている．具体的には，頂点 i $(0 \leq i < N)$ の色は $C[i]$ である．ある色が複数の頂点に塗られているかもしれないし，どの頂点にも塗られていない色が存在するかもしれない．スフィンクスの色が塗られた頂点は存在しない．すなわち，$0 \leq C[i] < N$ $(0 \leq i < N)$ が成り立つ．

パス $v_0, v_1, \ldots, v_k$ $(k \geq 0)$ はそのすべての頂点の色が等しいとき，すなわち $0 \leq l < k$ を満たすすべての l に対し $C[v_l] = C[v_{l+1}]$ が成り立つとき，**単色パス** であるという．さらに，頂点 p, q $(0 \leq p < N,$ $0 \leq q < N)$ が単色パスによって繋がれているとき，かつそのときに限り，頂点 p, q は同一の **単色成分** に属しているという．

あなたはどの頂点および辺が存在するかを知っているが，頂点の色は知らない．あなたは **塗り替え実験** を行うことで，頂点の色を知ろうとしている．

塗り替え実験において，あなたはいくつかの頂点を好きに選んでそれぞれ好きな色で塗り替えることができる．具体的には，塗り替え実験を行うためには，まず長さ N の整数列 E を選ぶ．E の各要素は -1 以上 N 以下でなければならない．次に，$0 \leq i < N$ を満たす各 i について，頂点 i が以下で定める色 $S[i]$ で塗り替えられる：

- $E[i] = -1$ ならば，$S[i] = C[i]$ である．すなわち，$S[i]$ は頂点 i の元の色である．
- そうでない場合，$S[i] = E[i]$ である．

塗り替え実験においては頂点をスフィンクスの色で塗ることができることに注意せよ．

そして，ギザの大スフィンクスは各頂点 i $(0 \leq i < N)$ を色 $S[i]$ で塗り替えた後のグラフに含まれる単色成分の個数をあなたに教える．この塗り替えはあくまで実験のためであり，**塗り替え実験の後，頂点の色は元に戻る**．

あなたの課題は，高々 $2\,750$ 回の塗り替え実験を行うことで，各頂点の色を特定することである．なお，すべての隣接する 2 頂点について，それらの色が等しいかどうかを正しく特定することができれば，部分点を獲得することが可能である．

実装の詳細

あなたは以下の関数を実装する必要がある．

```
std::vector<int> find_colours(int N,
    std::vector<int> X, std::vector<int> Y)
```

- N: グラフの頂点の数．
- X, Y: グラフの辺を表す長さ M の数列．
- この関数は，頂点の色を表す長さ N の数列 G を返さなければならない．
- この関数は各テストケースにおいてちょうど 1 度だけ呼び出される．

この関数は，塗り替え実験を行うために以下の関数を呼び出すことができる．

```
int perform_experiment(std::vector<int> E)
```

- E: 頂点がどのように塗り替えられるかを表す長さ N の数列．
- この関数は，塗り替え実験において整数列 E を選んで頂点を塗り替えた後の単色成分の個数を返す．
- この関数は最大で $2\,750$ 回まで呼び出すことができる．

採点プログラムは **適応的 (adaptive) ではない**．すなわち，各頂点の色は `find_colours` への呼び出しがなされる前から固定されている．

制約

- $2 \leq N \leq 250$
- $N - 1 \leq M \leq \frac{N(N-1)}{2}$
- $0 \leq X[j] < Y[j] < N$ $(0 \leq j < M)$
- $X[j] \neq X[k]$ または $Y[j] \neq Y[k]$ $(0 \leq j < k < M)$
- どの相異なる 2 つの頂点についても，それらを繋ぐパスが存在する．
- $0 \leq C[i] < N$ $(0 \leq i < N)$

小課題

小課題	得点	追加の制約
1	3	$N = 2$
2	7	$N \leq 50$
3	33	与えられるグラフはパスグラフである．すなわち，$M = N - 1$ であり，頂点 $j, j+1$ は隣接している $(0 \leq j < M)$．
4	21	与えられるグラフは完全グラフである．すなわち，$M = \frac{N(N-1)}{2}$ であり，どの相異なる 2 つの頂点も隣接している．
5	36	追加の制約はない．

それぞれの小課題において，あなたのプログラムがすべての隣接する 2 頂点についてそれらの色が等しいかどうかを正しく特定することができれば，部分点を獲得することが可能である．

より厳密には，小課題に含まれるすべてのテストケースにおいて `find_colours` が返した数列 G が C と全く同じ，すなわち $G[i] = C[i]$ $(0 \leq i < N)$ ならば，その小課題のすべての得点を獲得することができる．そうでなく，小課題に含まれるすべてのテストケースにおいて次の 2 つの条件が満たされるとき，その小課題の得点の 50% を獲得することができる．

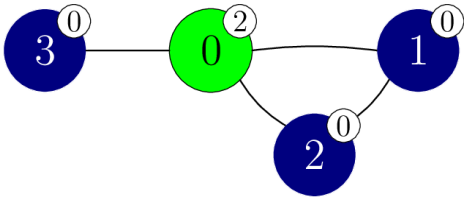
- $0 \leq G[i] < N$ $(0 \leq i < N)$
- $0 \leq j < M$ を満たすすべての j に対し，
  - $C[X[j]] = C[Y[j]]$ のとき，かつそのときに限り $G[X[j]] = G[Y[j]]$ である．

例

以下の呼び出しを考える．

```
find_colours(4, [0, 1, 0, 0], [1, 2, 2, 3])
```

この例において，（あなたには明かされていない）各頂点の色が $C = [2, 0, 0, 0]$ であるとする．このシナリオは下図のように表される．頂点の色は各頂点の右上に付けられた白いラベルによって表されている．



この関数は `perform_experiment` を以下のように呼び出すことができる．

```
perform_experiment([-1, -1, -1, -1])
```

この呼び出しにおいて，どの頂点も元の色を保っており，塗り替えられた頂点はない．

頂点 $1, 2$ について考える．いずれも色 0 で塗られているため，パス $1, 2$ は単色パスである．そのため，頂点 $1, 2$ は同一の単色成分に属する．

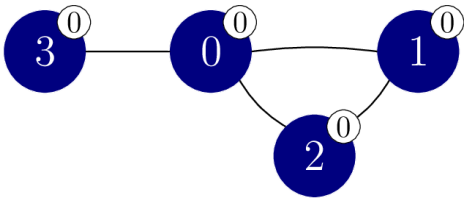
頂点 $1, 3$ について考える．いずれも色 0 で塗られているが，これらを繋ぐ単色パスは存在しないため，頂点 $1, 3$ は異なる単色成分に属する．

頂点の集合がそれぞれ $\{0\}, \{1, 2\}, \{3\}$ である 3 つの単色成分が存在するので，この呼び出しは 3 を返す．

次に `perform_experiment` を以下のように呼び出すことができる．

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

この呼び出しにおいて，頂点 0 が色 0 で塗り替えられ，下図のようになる．

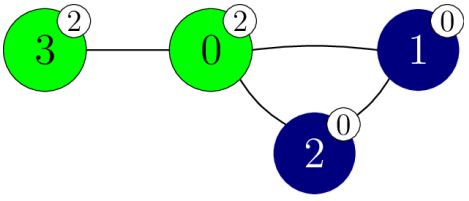


すべての頂点が同一の単色成分に属するため，この呼び出しは 1 を返す．これによって，頂点 $1, 2, 3$ の色が 0 であると導くことができる．

さらに，`perform_experiment` を以下のように呼び出すことができる．

```
perform_experiment([-1, -1, -1, 2])
```

この呼び出しにおいて，頂点 3 が色 2 で塗り替えられ，下図のようになる．



頂点の集合がそれぞれ $\{0, 3\}, \{1, 2\}$ である 2 つの単色成分が存在するので，この呼び出しは 2 を返す．これによって，頂点 0 の色が 2 であると導くことができる．

その結果，関数 `find_colours` は数列 $[2, 0, 0, 0]$ を返す．$C = [2, 0, 0, 0]$ であるので，仮にこのテストケースのみからなる小課題が存在した場合，小課題のすべての点数が与えられることになる．

$[1, 2, 2, 2]$ や $[1, 2, 2, 3]$ を返した場合でも小課題の 50% の点数が与えられることに注意せよ．

採点プログラムのサンプル

入力形式：

```
N  M
C[0]  C[1] ... C[N-1]
X[0]  Y[0]
X[1]  Y[1]
...
X[M-1]  Y[M-1]
```

出力形式：

```
L  Q
G[0]  G[1] ... G[L-1]
```

ここで，L は `find_colours` が返した数列 G の長さを表し，Q は `perform_experiment` の呼び出し回数を表す．