順列ゲーム (permgame)

Alice と Bob は幼馴染であり，知的なゲームを遊ぶことを愛している．今日，彼らはグラフ上における新たなゲームを遊んでいる．

そのゲームセットは 0 から $m-1$ までの番号が付けられている m 個の頂点と，0 から $e-1$ までの番号が付けられている e 本の辺からなる **連結な** グラフを含んでいる．i 番目の辺は頂点 $u[i]$ と頂点 $v[i]$ を結んでいる．

加えて，そのゲームセットは長さ n の順列 $p[0], p[1], \ldots, p[n-1]$ も含んでいる．ここで，$m \leq n$ である．順列は 0 から $n-1$ までの数が，ある順序でちょうど一度ずつ現れる数列である．順列 p の **スコア** は $p[i] = i$ となるインデックス i の個数である．

ゲームは最大で 10^{100} ターンまで続く．各ターンでは，以下の出来事が起こる：

1. もし Alice がゲームを終了すると決めた場合，ゲームは終了する．
2. そうでない場合，Alice は **相異なるインデックス** $t[0], t[1], \ldots, t[m-1]$ を選ぶ．ただし，$0 \leq t[i] < n$ を満たす必要がある．ゲームでは $t[0] < t[1] < \ldots < t[m-1]$ であることは **要求されない** ことに注意せよ．
3. Bob は $0 \leq j < e$ を満たすグラフの辺のインデックスを選び，$p[t[u[j]]]$ と $p[t[v[j]]]$ を交換する．

Alice は最終的な順列のスコアを最大化しようとし，Bob は最終的な順列のスコアを最小化しようとしている．

あなたの課題は Alice の手助けをして Bob とゲームを行うことである．ただし，Bob の行動は採点プログラムによりシミュレーションされる．

最適なスコアとは，Alice と Bob が最適に行動した場合の最終的な順列のスコアである．

あなたは最適なスコアを求め，さらに，ゲームの終了時にその値以上のスコアを達成するように Bob とゲームを行う必要がある．

Bob がどのような行動をとった場合でも Alice の戦略は正しく機能する必要がある．Bob は最適でない行動をとる可能性があることに注意せよ．

実装の詳細

あなたは以下の関数を実装する必要がある：

```
int Alice(int m, int e, std::vector<int> u, std::vector<int> v,
          int n, std::vector<int> p)
```

- m：グラフの頂点の個数
- e：グラフの辺の本数
- u, v：グラフの辺を表す長さ e の配列
- n：順列の長さ
- p：順列を表現する長さ n の配列
- この関数はちょうど 1 回呼び出される．
- この関数はゲームの最適なスコアである整数を返さなければならない．

この関数の中で，あなたは以下の関数を呼び出すことができる：

```
int Bob(std::vector<int> t)
```

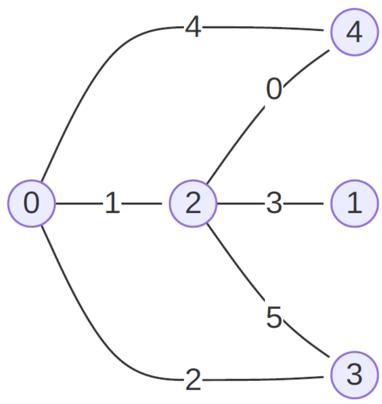
- t：相異なるインデックスを含んだ長さ m の配列．ただし，各 $i \neq j$ に対し，$0 \leq t[i] < n$ および $t[i] \neq t[j]$ を満たすこと．
- この関数は $0 \leq j < e$ を満たす 1 つの整数 j を返す．
- この関数は何回でも呼び出すことができる．

例

以下の呼び出しを考える：

```
Alice(5, 6, [4, 0, 3, 1, 4, 2], [2, 2, 0, 2, 0, 3],
      10, [8, 2, 7, 6, 1, 5, 0, 9, 3, 4])
```

グラフは以下のように表される：



また，p は最初 $[8, 2, 7, 6, 1, 5, 0, 9, 3, 4]$ である．

これらの制約の下で，最適なスコアは 1 となることが証明できる．

Alice が以下の 4 回の行動をしたと考える：

`Bob` の引数となる t	`Bob` の返り値	対応する p のインデックス	Bob が交換した後の p
$[3,1,5,2,0]$	5	$5,2$	$[8,2,5,6,1,7,0,9,3,4]$
$[9,3,7,2,1]$	0	$1,7$	$[8,9,5,6,1,7,0,2,3,4]$
$[5,6,7,8,9]$	1	$5,7$	$[8,9,5,6,1,2,0,7,3,4]$
$[7,5,2,3,6]$	3	$5,2$	$[8,9,2,6,1,5,0,7,3,4]$

上にあげた Alice と Bob の行動は必ずしも最適なものであるとは限らないことに注意せよ．これらの行動は単に説明のためのものである．また，順列の最初のスコアは 1 であるため Alice はただちにゲームを終了させることもできる．

Alice が上の全ての行動をした後，順列のスコアは 3 となる ($p[2]=2, p[5]=5, p[7]=7$)．

最後に，関数 `Alice()` は最適なスコアである 1 を返す．

もし Alice が Bob とゲームを行うことでスコアを 3 にすることが達成できた場合でも，`Alice()` の返り値を 1 の代わりに 3 とした場合，あなたは得点を得ることができない．

制約

- $2 \leq m \leq 400$
- $m-1 \leq e \leq 400$
- $0 \leq u[i], v[i] < m$
- $m \leq n \leq 400$
- $0 \leq p[i] < n$
- グラフは連結で，自己ループや多重辺を含まない．
- p は順列である．すなわち，任意の $i \neq j$ について $p[i] \neq p[j]$．

小課題

1. (6 点) $m=2$
2. (6 点) $e>m$
3. (10 点) $e=m-1$
4. (24 点) $e=m=3$
5. (24 点) $e=m=4$
6. (30 点) $e=m$

各小課題について，あなたは部分点を獲得することができる．k をターンの回数 (すなわち，`Bob()` の呼び出し回数) として，r を小課題の全てのテストケースの中における $\frac{k}{n}$ の最大値とする．このとき，その小課題におけるあなたの点数は，小課題の配点に以下の値を掛けたものである．

条件	かけられる数
$12 \leq r$	0
$3 < r < 12$	$1 - \log_{10}(r-2)$
$r \leq 3$	1

特に，あなたが $3n$ ターン以内で問題を解けば，あなたはその小課題について満点が得られる．$12n$ ターン以上を要すれば，その小課題は 0 点となる (`Output isn't correct` と表示される)．

採点プログラムのサンプル

採点プログラムのサンプルは以下の形式で入力を受け取る：

- 1 行目：m e
- $2+i$ $(0 \leq i \leq e-1)$ 行目：$u[i]$ $v[i]$
- $2+e$ 行目：n
- $3+e$ 行目：$p[0]$ $p[1]$ $\ldots$ $p[n-1]$

採点プログラムのサンプルは以下の形式で出力する：

- 1 行目：最終的な順列 p
- 2 行目：`Alice()` の返り値
- 3 行目：最終的な順列のスコア
- 4 行目：ターンの回数