最長の旅 (Longest Trip)

IOI 2023 の運営は困難に面している．後日予定された Ópusztaszer への旅の計画を作り忘れてしまったのだ．ただ，おそらくまだ遅すぎはしないはずだ...

Ópusztaszer には N 個のランドマークがあり，0 から $N-1$ までの番号が付けられている．いくつかのランドマークの組は*双方向に移動可能な* **道路** で繋がれている．どのランドマークの組についても，それらの間をつなぐ道路は最大 1 本である． IOI 2023 の運営は，どのランドマークの組が道路で繋がれているかを*知らない*.

Ópusztaszer の道路網の **密度** は，次のとき δ **以上**であると言う：すべての相異なるランドマークの 3 つ組の間の道路は δ 本以上である．言い換えると，任意のランドマークの 3 つ組 (u, v, w) $(0 \leq u < v < w < N)$ について，ランドマークの組 $(u, v), (v, w), (u, w)$ のうち少なくとも δ 組は道路で繋がれている．

IOI 2023 の運営は，道路網の密度がある正整数 D 以上であるとわかっている．ここで，D は 3 以下であることに注意すること．

IOI 2023 の運営は，いくつかのランドマーク間の道路網に関する情報を入手するため，Ópusztaszer の運行管理者に **電話** をかけることができる．それぞれの電話において，2 つの空でないランドマークの配列 $[A[0], \ldots, A[P-1]]$ と $[B[0], \ldots, B[R-1]]$ を指定する必要がある．ランドマークは相異なる必要がある．すなわち，以下が満たされていなければならない．

- i, j $(0 \leq i < j < P)$ について，$A[i] \neq A[j]$
- i, j $(0 \leq i < j < R)$ について，$B[i] \neq B[j]$
- i, j $(0 \leq i < P, 0 \leq j < R)$ について，$A[i] \neq B[j]$

それぞれの電話において，運行管理者は配列 A に含まれるランドマークと配列 B に含まれるランドマークを繋ぐ道路があるかどうかを返答する．より正確には，運行管理者は $0 \leq i < P, 0 \leq j < R$ を満たす i, j の組を調べる．$A[i]$ と $B[j]$ が道路で繋がれているような i, j の組が存在する場合，運行管理者は `true` と返答する．それ以外の場合，運行管理者は `false` と返答する．

長さ l の **旅** とは，*相異なる* ランドマークの配列 $t[0], t[1], \ldots, t[l-1]$ であって，各 i $(0 \leq i \leq l-2)$ についてランドマーク $t[i]$ とランドマーク $t[i+1]$ が道路で繋がれているようなものである．ある長さ l の旅について，長さ $l+1$ 以上の長さの旅が存在しないとき，その旅を**最長の旅**と呼ぶ.

あなたの課題は，運行管理者に電話をかけることで，Ópusztaszer での最長の旅を見つけることである．

実装の詳細

あなたは以下の関数を実装する必要がある．

```
int[] longest_trip(int N, int D)
```

- N：Ópusztaszer におけるランドマークの数．
- D：道路網の密度として保証される下限．
- この関数は最長の旅を表す配列 $t = [t[0], t[1], \ldots, t[l-1]]$ を返す必要がある．
- この関数はそれぞれのテストケースについて **複数回** 呼び出される可能性がある．

上の関数内では以下の関数を呼び出すことができる．

```
bool are_connected(int[] A, int[] B)
```

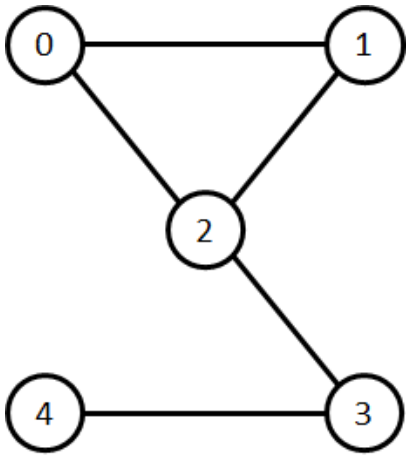
- A：相異なるランドマークの空でない配列．
- B：相異なるランドマークの空でない配列．
- A と B は同じ要素を含んではならない．
- 配列 A に含まれるランドマークと配列 B に含まれるランドマークを繋ぐ道路が存在する場合，`true` を返す．存在しない場合，`false` を返す．
- この関数は `longest_trip` の各呼び出し内で最大 $32\,640$ 回呼び出すことができ，合計では最大 $150\,000$ 回呼び出すことができる．
- この関数の呼び出し全体を通して，この関数に渡される配列 A, B の長さの合計は $1\,500\,000$ を超えてはならない．

採点プログラムは **適応的 (adaptive) ではない**．それぞれの提出は同じテストケースのセットで採点される．すなわち，N, D の値や道路で繋がれたランドマークの組は，`longest_trip` が呼び出される前に固定されている．

例

例 1

$N = 5, D = 1$ であり，道路網が下図の通りになっているシナリオを考えよう．



`longest_trip` は以下のように呼び出される．

```
longest_trip(5, 1)
```

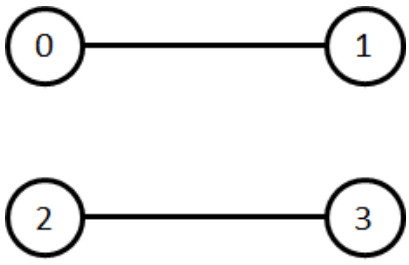
この関数は以下のように `are_connected` を呼び出すことができる．

呼び出し	道路で繋がれたランドマークの組	戻り値
`are_connected([0], [1, 2, 4, 3])`	$(0,1), (0,2)$	`true`
`are_connected([2], [0])`	$(2,0)$	`true`
`are_connected([2], [3])`	$(2,3)$	`true`
`are_connected([1, 0], [4, 3])`	なし	`false`

4 回目の呼び出しの後，ランドマークの組 $(1,4), (0,4), (1,3), (0,3)$ は *いずれも道路で繋がれていない* ことがわかる．この道路網の密度が $D = 1$ 以上であることより，ランドマークの 3 つ組 $(0,3,4)$ について考えると，組 $(3,4)$ は道路で繋がれている必要があることがわかる． 同様にして，ランドマーク 0 とランドマーク 1 は道路で繋がれている必要があることがわかる．

この時点で，$t = [1,0,2,3,4]$ は長さ 5 の旅であり，長さが 5 より長い旅は存在しないと結論づけられる．したがって，`longest_trip` は `[1, 0, 2, 3, 4]` を返すことができる．

異なるシナリオとして，$N = 4, D = 1$ であり，道路網が下図の通りになっている状況を考えよう．



`longest_trip` は以下のように呼び出される．

```
longest_trip(4, 1)
```

このシナリオにおいて，最長の旅の長さは 2 である．したがって，are_connected を数回呼び出した後，longest_trip は $[0, 1], [1, 0], [2, 3], [3, 2]$ のうちのどれかを返せばよい．

例 2

小課題 0 は $N = 256$ であるテストケースの例を含んでいる． このテストケースはコンテストシステムからダウンロードできる添付されたパッケージに含まれている．

制約

- $3 \leq N \leq 256$
- longest_trip の呼び出しの全体を通して，N の総和は $1\,024$ を超えない．
- $1 \leq D \leq 3$

小課題

1. (5 点) $D = 3$
2. (10 点) $D = 2$
3. (25 点) $D = 1$．$l^\star$ を最長の旅の長さとする．longest_trip は 長さ $l^\star$ の旅を表す配列を返す必要はない．その代わりに，長さが $\left\lceil \frac{l^\star}{2} \right\rceil$ 以上の旅を表す配列を返す必要がある．
4. (60 点) $D = 1$

小課題 4 において，あなたの得点は longest_trip の 1 回の呼び出しにおける are_connected の呼び出し回数に基づいて決定される． この小課題に対するあなたの点は以下の表に従って計算される．

条件	点
$2\,750 < q \leq 32\,640$	20
$550 < q \leq 2\,750$	30
$400 < q \leq 550$	45
$q \leq 400$	60

各テストケースについて，are_connected の呼び出しが制約を満たしていない場合，もしくは longest_trip の返す配列が正しくない場合，あなたの解法のその小課題に対する得点は 0 点となる．

採点プログラムのサンプル

C をシナリオの数，すなわち longest_trip の呼び出し回数とする． 採点プログラムのサンプルは以下の形式で入力を読み込む．

- 1 行目：C

C 個のシナリオの説明が以下に続く．

採点プログラムのサンプルは各シナリオについて以下の形式で入力を読み込む．

- 1 行目：N D
- $1+i$ 行目 $(1 \le i < N)$：$U_i[0]$ $U_i[1]$ $\ldots$ $U_i[i-1]$

ここで，各 U_i $(1 \le i < N)$ は長さ i の配列であり，どのランドマークのペアが道路で繋がれているかを表している． 各 i, j $(1 \le i < N, 0 \le j < i)$ について，

- ランドマーク j とランドマーク i が道路で繋がれている場合，$U_i[j]$ は 1 となる．
- ランドマーク j とランドマーク i が道路で繋がれていない場合，$U_i[j]$ は 0 となる．

各シナリオにおいて，longest_trip が呼び出される前に，採点プログラムのサンプルは道路網の密度が D 以上であるかを確認する．

採点プログラムのサンプルがプロトコルの違反を検出した場合，採点プログラムの出力は `Protocol Violation: <MSG>` となる．ここで，<MSG> は以下のエラーメッセージのうちのいずれかである．

- `invalid array`：are_connected の呼び出しにおいて，配列 A, B のうち少なくとも片方が
  - 空である．
  - 0 以上 $N-1$ 以下でない整数の要素を含んでいる．
  - 少なくとも 2 回同じ要素を含んでいる．
- `non-disjoint arrays`：are_connected の呼び出しにおいて，配列 A, B が同じ要素を含んでいる．
- `too many calls`：longest_trip の現在の呼び出しにおける are_connected の呼び出し回数が $32\,640$ 回を超えている，または合計で $150\,000$ 回を超えている．
- `too many elements`：全体を通じて are_connected に渡されたランドマークの数の合計が $1\,500\,000$ を超えている．

そして，あるシナリオに対する longest_trip の戻り値が，非負整数 l を用いて $t[0], t[1], \ldots, t[l-1]$ であるとすると，採点プログラムのサンプルはこのシナリオに対して，以下の形式で 3 行出力する．

- 1 行目：l
- 2 行目：$t[0]$ $t[1]$ $\ldots$ $t[l-1]$
- 3 行目：このシナリオにおける are_connected の呼び出し回数

最後に，採点プログラムのサンプルは以下のように出力する．

- $1 + 3 \cdot C$ 行目：すべての longest_trip の呼び出しにおける are_connected の呼び出し回数の最大値