Test č. 2, úloha č. 1

V priloženej kostre nájdete známu implementáciu balíka `graphs` z prednášky. Súvislý graf G nazývame *bipartitný*, ak sa jeho množina vrcholov $V(G)$ dá rozdeliť na dve množiny A a B také, že $A \cap B = \emptyset$ a zároveň $A \cup B = V(G)$ pre ktoré platí, že pre ľubovoľné vrcholy $a_1, a_2 \in A, b_1, b_2 \in B$ neexistujú v grafe hrany a_1a_2 a b_1b_2. Všetky hrany grafu sú teda iba medzi množinami A a B. Množiny A a B nazývame *bipartície*. Nesúvislý graf je bipartitný, ak je každý jeho komponent bipartitý. Doplňte implementáciu verejnej triedy `Bipartite` implementujúcej statické metódy

- `public static Set<Integer> getBipartition(UndirectedGraph graph, int vertex)`, ktorá pre vrchol `vertex` v neorientovanom grafe `graph` nájde a vráti jeho bipartíciu a vráti ju ako množinu `Set<Integer>`, za predpokladu, že existuje. Pokiaľ neexistuje (pretože komponent súvislosti, v ktorom sa `vertex` nachádza nie je bipartitný), vráti metóda `null`.
- `public static boolean isBipartite(UndirectedGraph graph)`, ktorá pre graf vráti `true` práve vtedy ak je graf bipartitný (ak je každý jeho komponent bipartitný). Inými slovami, vráti `true` práve vtedy, ak pre každý vrchol existuje bipartícia, v ktorej sa daný vrchol nachádza.

Na testovač odovzdajte iba súbor `Bipartite.java`. V priloženej kostre máte aj triedu `Testovac`, ktorá obsahuje funkciu na načítanie grafu. Využite ju na otestovanie svojho riešenia. Môžete využiť príklady nižšie.

Príklad 1 (graf na 3 vrcholoch bez hrán):

```
Bipartite.getBipartition(g,0)  // == [0]
Bipartite.getBipartition(g,1)  // == [1]
Bipartite.getBipartition(g,2)  // == [2]
Bipartite.isBipartite(g)       // == true
```

Vstup pre načítanie:

```
LISTS
3 0
```

Príklad 2 (graf je trojuholník):

```
Bipartite.getBipartition(g,0)  // == null
Bipartite.getBipartition(g,1)  // == null
Bipartite.getBipartition(g,2)  // == null
Bipartite.isBipartite(g)       // == false
```

Vstup pre načítanie:

```
MATRIX
3 3
0 1
0 2
1 2
```

Príklad 3 (graf je štvoruholník):

```
Bipartite.getBipartition(g,0)  // == [0, 2]
Bipartite.getBipartition(g,1)  // == [1, 3]
Bipartite.isBipartite(g)       // == true
```

Vstup pre načítanie:

```
MATRIX
4 4
0 1
1 2
2 3
0 3
```