

Problema das duplas

Considere uma turma da Ufes contendo um número **par** de alunos n . O professor desta turma passou um trabalho para ser feito em dupla, e ele gostaria de otimizar a satisfação de todos os alunos com a sua respectiva dupla. Para tal, o professor pediu para cada aluno construir uma lista com as suas preferências na formação da dupla. A lista de um aluno contém todos os demais colegas da turma e está ordenada de forma decrescente por preferências, não havendo empate de preferência entre as pessoas da lista.

Ajude o professor criando um programa que recebe como entrada as listas de preferência dos n alunos e exibe como saída as $n/2$ duplas que melhor atendem aos critérios dos alunos, garantindo que nenhum estudante vai querer trocar de dupla. Em outras palavras, a sua configuração de duplas deve ser *estável*, isto é, não podem haver dois alunos que não formam uma dupla, mas que possuem uma preferência mútua maior que as dos seus pares atuais.

Exemplo 1:

Os alunos são identificados por números inteiros na faixa entre 0 e $n-1$. Considere o exemplo abaixo com $n = 4$.

Aluno: 0	Preferências: 1 2 3
Aluno: 1	Preferências: 0 2 3
Aluno: 2	Preferências: 3 1 0
Aluno: 3	Preferências: 2 1 0

Figura 1: Exemplificação da lista de preferência 1.

O Aluno 0 tem como maior preferência o Aluno 1 e vice-versa. Assim, de imediato temos que uma dupla com estes alunos deve ser formada. O mesmo interesse mútuo ocorre entre os Alunos 2 e 3. Assim, as duplas que devem ser formadas são (0, 1) e (2, 3).

Exemplo 2:

É essencial destacar que é possível que um conjunto de escolhas dos alunos não admita uma solução. Considere uma variação do exemplo anterior como abaixo.

Aluno: 0	Preferências: 1 2 3
Aluno: 1	Preferências: 2 0 3
Aluno: 2	Preferências: 0 1 3
Aluno: 3	Preferências: 0 1 2

Figura 2: Exemplificação da lista de preferência 2.

Para este exemplo, qualquer solução vai exigir que 3 fique pareado com 0, 1, ou 2, mas todos estes alunos preferem qualquer outro colega a 3. Considere as duplas (0,3) e (1,2). Uma configuração mais favorável para os Alunos 0 e 2 seria formar a dupla (0, 2), as isto deixaria a outra dupla como (1, 3), novamente criando o mesmo problema. Não há uma solução estável para este exemplo. Neste caso, a saída do programa dever ser somente o valor -1, indicando que não há solução.

Entrada

A entrada é um arquivo contendo na primeira linha o valor n . A seguir, as demais n linhas contém as preferências dos alunos na ordem de 0 a $n-1$. Cada linha contém as $n-1$ preferências de um aluno separadas por um espaço.

Saída

A saída deve ser o identificador da dupla de cada aluno, na ordem de 0 a $n-1$, com um aluno por linha, ou -1, indicando que não há solução.

OBS: É proibido utilizar a biblioteca *matching* de Python.

Exemplo

Entrada	Saída
<pre> 4 1 2 3 0 2 3 3 1 0 2 1 0 </pre>	<pre> 1 0 3 2 </pre>
<pre> 4 1 2 3 1 2 0 3 2 0 1 3 3 0 1 2 </pre>	<pre> -1 </pre>
<pre> 10 8 2 1 3 5 4 9 7 6 6 4 7 3 8 5 9 2 0 9 7 5 8 6 1 3 0 4 1 6 5 4 2 7 0 9 8 5 1 7 6 3 8 9 2 0 1 9 0 2 4 6 7 3 8 9 3 1 5 4 7 0 8 2 9 3 1 4 2 6 0 5 8 3 9 4 5 1 6 0 7 2 0 2 3 7 1 4 6 5 8 </pre>	<pre> 8 7 9 6 5 4 3 1 0 2 </pre>

11 $\{ [0, 2, 3], [1, 2, 3], [3, 1, 0], [2, 1, 0] \}$

$[0, 2]$

$[1, 0, 1]$

$[1, 2, 3],$

$[1, 2, 3],$

$[1, 2, 3], [2, 0, 3], [2, 3, 3]$

$[0, 4, 2]$

$x = 1, [0, 2, 3]$
 $is(x) in is[3]$

$[1, 0, 1]$