

VII BXComp

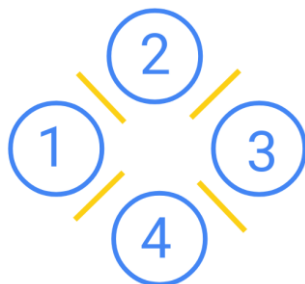
7º Campeonato de Programação para Calouros do Curso de Sistemas de Informação 2017

3ª Etapa – Desafio 3

Filósofos

Durante a Semana da Ciência realizada na Escola de Artes Ciências e Humanidades, foram convidados filósofos para participar de um debate sobre mitologia grega. Após o debate, os filósofos foram convidados para participar de uma outra atração que estava ocorrendo no evento: degustação da gastronomia chinesa.

Os filósofos aceitaram o convite e foram instruídos a se acomodarem em uma mesa circular e aguardarem para serem servidos. Porém, houve um engano e apenas metade dos hashis foram entregues para as refeições. Ou seja, havia apenas um hashi por filósofo. Nesse cenário, um filósofo comerá apenas se os filósofos adjacentes não estão comendo, como descrito abaixo



No caos resultante, para tentar solucionar esse problema, foram criadas regras e um sistema no qual é distribuída aos filósofos a oportunidade de tentar obter os hashis e comer.

Tarefa

Sua tarefa é implementar um programa que determine quais filósofos terminaram suas refeições ao fim do dia, dada a sequência aleatória de oportunidades distribuídas. Os **F** filósofos foram enumerados e pretendem realizar, individualmente, **X** refeições, divididas em **Y** porções que levam um instante para serem comidas. Para cada instante, o sistema dá oportunidades para um filósofo específico que se comporta da seguinte maneira:

Se é dada a oportunidade a um filósofo que não tem hashis, ele tenta obter os hashis. Para isso, os filósofos adjacentes não podem estar comendo. Se ele obtém os hashis ele imediatamente come uma porção e não larga os hashis até terminar as porções da refeição atual. Se ele não obtém ele apenas filosofa. Se é dada a oportunidade a um filósofo que já tem hashis ele a ignora e continua comendo. Ele também ignora a oportunidade se ele já comeu todas as porções de todas as suas refeições.

A cada instante, todos os filósofos que possuem um par de hashis comem uma porção. Quando um filósofo termina uma refeição ele larga os hashis e passa o próximo instante filosofando, portanto ignorará uma oportunidade se a mesma for dada no instante seguinte.

Entrada

A entrada conterá **T** casos de teste, sendo que **T** é um inteiro positivo dado na primeira linha de entrada. Para cada caso, haverá um inteiro **F** ($1 < F \leq 10$) em uma linha, que é a quantidade de filósofos à mesa. Na linha seguinte, haverá **F** pares de inteiros **X** ($1 \leq X \leq 10$) e **Y** ($1 \leq Y \leq 10$) separados por espaços representando o número de refeições e o número de porções por refeição, respectivamente de cada filósofo. A linha seguinte conterá um número arbitrário de inteiros **O** ($1 \leq O \leq F$) que representa o filósofo que recebeu a oportunidade no instante atual.

Saída

Para cada caso de teste, seu programa deve imprimir, se houver um ou mais filósofos que terminaram todas as porções de todas suas refeições, uma linha que contém “Filósofos alimentados: ” e os números dos filósofos alimentados em ordem crescente, separados por espaços entre si e sem espaço após o último filósofo. Caso nenhum filósofo tenha terminado a refeição, seu programa deve imprimir a mensagem “Todos os filósofos estão famintos”.

Exemplo de Entrada

```
3
3
2 2 3 1 1 7
1 1 1 1 1 1 2 3 2 1 1 1 1 1
4
3 2 2 3 2 2 1 7
1 1 2 3 1 3 4 4 4 4
2
2 1 2 1
1 2 1 2
```

Exemplo de Saída

```
Filósofos alimentados: 1 3
Todos os filósofos estão famintos
Filósofos alimentados: 1 2
```