Instituto de Computação - Unicamp

MC102 - Algoritmos e Programação de Computadores

Laboratório 04 - Balanceamento de Carga na Aliança Rebelde

Prazo de entrega: **05/04/2019 23:59:59**

Peso: **1**

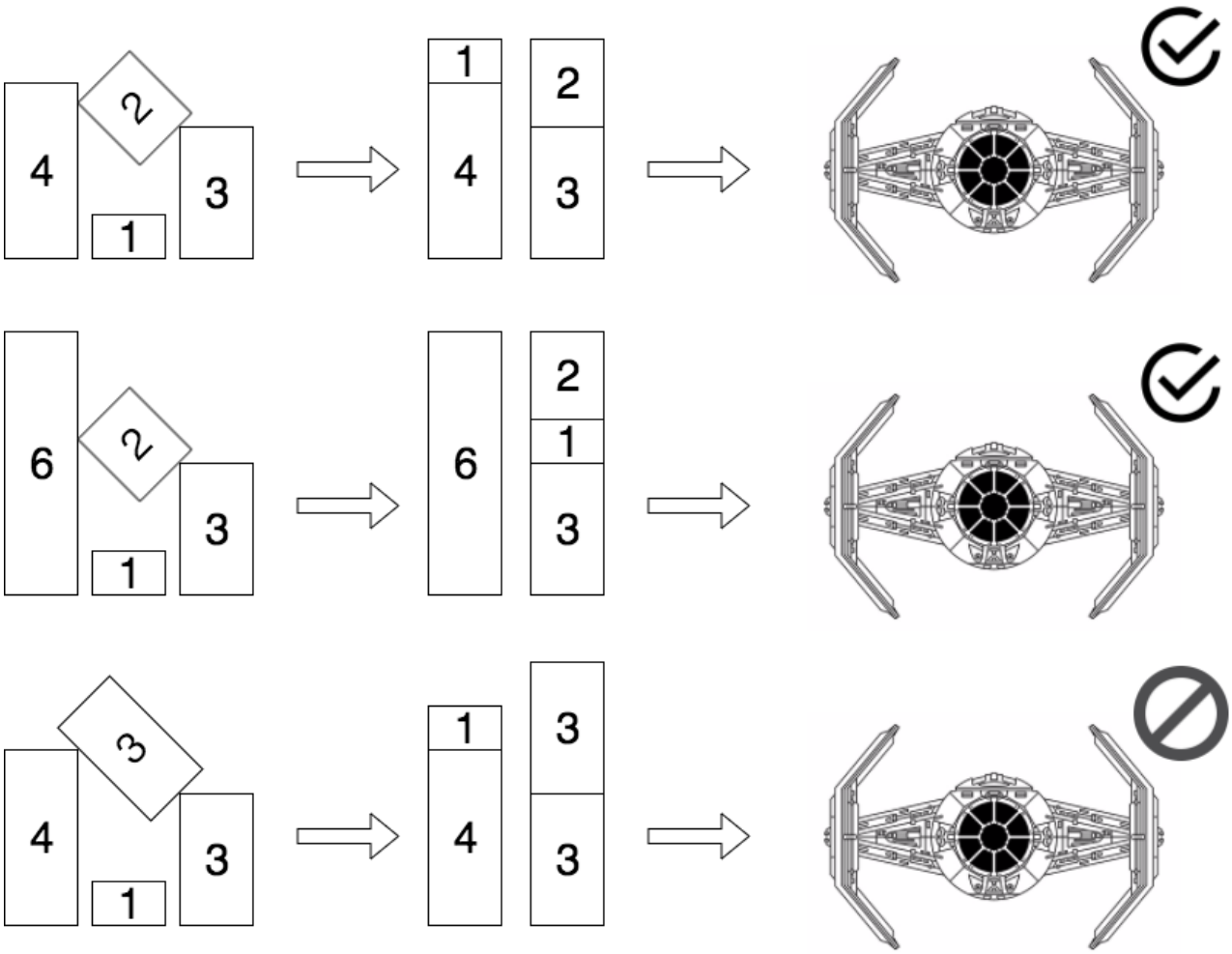
Professor: Eduardo C. Xavier
Professor: Luiz F. Bittencourt

Descrição

O transporte interestelar de cargas apresenta uma série de medidas de segurança, e uma dessas medidas é a distribuição de pesos no interior da espaçonave. Essa medida estabelece que deve haver um balanceamento na carga que será transportada, ou seja, o peso na parte traseira da aeronave deve ser compatível com o peso na parte da frente da aeronave. Quando o balanceamento é bem realizado, a espaçonave voa mais rápido e consome menos combustível.
Nas bases de abastecimento da aliança rebelde cada espaçonave é carregada com 4 containers. Esses containers estão carregados com munições e suprimentos. Cada container apresenta seu peso total em toneladas, representado por um número inteiro positivo.
Exemplo:
c_1 = 4 toneladas.
c_2 = 7 toneladas.
c_3 = 3 toneladas.
c_4 = 8 toneladas.

Objetivo

Como engenheiro de software da aliança rebelde, você deve criar um programa que, dado o peso de 4 containers, determina se é possível estabelecer um balanceamento da carga perfeito, ou seja, se é possível dividir a carga na parte traseira e dianteira da espaçonave de tal forma que o peso na parte traseira seja igual ao peso na parte dianteira.



Nos dois primeiros exemplos acima vemos que é possível separar os containers em 2 grupos, tal que a soma de pesos do grupo à esquerda é igual ao da direita, indicando que é possível distribuir os containers desta forma na espaçonave. No último exemplo acima, é impossível criar 2 grupos com os pesos 1, 3, 3, 4 tal que os dois grupos tenham pesos iguais.

Entrada

A entrada consiste de 4 números inteiros positivos, um por linha, contendo respectivamente, os valores de c_1, c_2, c_3 e c_4.

Saída

A saída deverá conter apenas uma linha, contendo somente `sim` ou `nao`. Caso seja possível obter um balanceamento perfeito da carga a saída deverá ser `sim`, caso contrário, a saída deverá ser `nao`.

Exemplos

Teste 01

Entrada

```
31
31
19
43
```

Saída

```
sim
```

Teste 04

Entrada

```
31
21
16
34
```

Saída

```
nao
```

Teste 12

Entrada

```
105
32
```

```
35
38
```

Saída

```
sim
```

Para mais exemplos, consulte os testes abertos no Susy.

Observações

- O número máximo de submissões é **10**;
- O seu programa deve estar completamente contido em um único arquivo denominado `lab04.py`.
- Para a realização dos testes do SuSy, a compilação dos programas desenvolvidos em C irá considerar o comando:
  `python3 lab04.py`;
- Você deve incluir, no início do seu programa, uma breve descrição dos objetivos do programa, da entrada e da saída, além do seu nome e do seu RA;
- Indente corretamente o seu código e inclua comentários no decorrer do seu programa.

Critérios importantes

Independentemente dos resultados dos testes do SuSy, o não cumprimento dos critérios abaixo implicará em nota zero nesta tarefa de laboratório.