

1 Introdução

Montanha Russa. Suponha que existam n passageiros e um carro em uma montanha russa. Os passageiros, repetidamente, esperam para dar uma volta no carro. O carro tem capacidade para C passageiros, com $C < n$. O carro só pode partir quando estiver cheio. Após dar uma volta cada passageiro passeia pelo parque de diversões até voltar para a próxima volta. Você deve simular k voltas do carro.

Tanto o carro como os passageiros devem ser representados por tarefas.

Os processos passageiro executam o seguinte código:

```
process passageiro {
  while (!fechouParque) {
    entraNoCarro();
    saiuDoCarro();
    passeiaPeloParque(); // tempo variável
  }
}
```

Processos carro:

```
process carro {
  while (existemPassageirosNoParque) {
    esperaEncher();
    daUmaVolta();
    esperaEsvaziar();
    volta++; // serve como parâmetro para fechar o parque
  }
}
```

2 Requisitos

Novamente, a parte acima deverá servir apenas como base para começar a escrever o EP. Para critério de nota serão considerados os seguintes requisitos:

- O programa deve ser extensível para mais de um carro, neste caso os carros não devem poder ultrapassar (isto deve ser controlado apenas no procedimento de chegada ao desembarque dos carros);
- Caso a fila de embarque esteja vazia, e pelo menos um passageiro a bordo, o carro deve partir mesmo que não estiver cheio;
- As tarefas passageiro devem ter tamanhos diferentes (isto é, devem existir passageiros que representem entre 1 e C passageiros). Logo, para completar o carro deve ser usada uma política de preenchimento - passageiros “menores” podem furar a fila (proponha e implemente uma política justa);
- Interface gráfica desejável, mas saída texto bem organizada também serve.

3 Regras

O EP pode ser feito em pares (desde que os dois trabalhem), e a linguagem de programação é livre. Desta vez não serão abertas exceções para trios.