Suponha um parque com um carro de montanha-russa e N visitantes. Os visitantes repetidamente esperam para andar no carro, que pode conter no máximo C passageiros, onde C < N. Porém, o carro só pode iniciar uma volta quando estiver cheio. Depois de terminar uma volta, cada passageiro sai do carro e volta para a fila do brinquedo.

Implemente um monitor para esse problema e também a thread carro e passageiros. A solução deve cumprir os seguintes requisitos obrigatórios:

* O carro sempre inicia uma volta com exatamente C passageiros;
* Nenhum passageiro pode pular fora do carro enquanto o carro está dando uma volta;
* Nenhum passageiro saltará para dentro do carro enquanto o carro está dando uma volta;
* Nenhum passageiro vai permanecer no carro após o término de uma volta (tentar dar duas voltas pagando apenas 1 bilhete);
* Os passageiros da volta n só poderão começar a entrar no carro depois que todos os passageiros da volta n-1 já saíram do carro.

Veja abaixo um esboço para as threads que definem o comportamento do carro e dos passageiros:

thread carro:

for(int i=0; i < n; ++i) {

montanharussa.espera\_encher();

/\* dá volta \*/

montanharussa.espera\_esvaziar();

}

thread cliente:

for(int i=0; i < n; ++i) {

montanharussa.entra\_no\_carro();

montanharussa.sai\_do\_carro();

/\* passeia no parque \*/

}