

Considere um estacionamento onde existam N vagas de veículos, nesse estacionamento existem K entradas e M saídas e um conjunto de P painéis que divulgam a quantidade de vagas ao público. Sua tarefa será simular o controle de vagas do estacionamento, evitando que veículos entrem no estacionamento sem que existam vagas disponíveis.

Sua implementação deve apresentar os seguintes elementos:

- Quantidade de vagas (N): variável global que representa a quantidade de vagas disponíveis em um determinado momento;
- Entrada: rotina que atualiza a quantidade de vagas disponíveis (decrementa), se a quantidade de vagas for igual a zero, o veículo não poderá entrar no estacionamento, nesse caso a entrada ficará bloqueada. A atualização da quantidade de vagas exige acesso exclusivo (exclusão mútua).
- Saída: rotina que atualiza a quantidade de vagas disponíveis (incrementa), desbloqueia uma entrada bloqueada (caso exista) por falta de vaga. A atualização da quantidade de vagas exige acesso exclusivo (exclusão mútua).
- Painel: rotina (thread) que realiza consultas sobre o número de vagas disponíveis no estacionamento (e escreve na tela esse valor). Existem um conjunto de painéis (a quantidade deve ser definida por uma constante no código). Sua implementação deve permitir o acesso de vários painéis simultaneamente para aquisição da quantidade de vagas disponíveis.
- Carros: representado por uma thread, sua implementação deve criar um determinado número de carros. Deve apresentar a seguinte estrutura:

```
while (true)
{
    Entrada; // entrar no estacionamento
    // aguardar tempo aleatório
    // imprimir mensagem: Carro <X> dentro do estacionamento
    Saída; // sair do estacionamento
    // imprimir mensagem: Carro <X> fora do estacionamento
}
```

Sua solução deve ser implementada com o uso de semáforos. Utilizar o menor número possível.