

A5 – Pilhas

Problema: o estacionamento do Senhor EDCarlos

A cidade EDCity possui estacionamentos inusitados. O senhor EDCarlos Alkaline é dono de um destes estacionamentos. O estacionamento é composto de duas pilhas com tamanho máximo de 10 carros. Assim, existem dois estacionamentos: o principal e o secundário. O principal armazena os carros enquanto o secundário é utilizado apenas para operacionalização do estacionamento principal. Portanto, os carros vão chegando e sendo inseridos no estacionamento principal, caso haja vagas. Para recuperar um carro, o dono informa o id (inteiro) do carro (que é único) e o Sr EDCarlos desempilha os carros, empilhando-os no estacionamento secundário até que se ache o carro do indivíduo. Ao final, os carros são movidos de volta (empilhados) no estacionamento principal.

Faça um programa em C para simular o estacionamento do Sr. EDCarlos. Assuma que os ids dos carros são únicos. A cada chamada da função pop do estacionamento principal, deve-se imprimir o id do carro que está sendo movido. Se o id para remoção não for encontrado, deve-se emitir a mensagem “underflow”, quando tentar remover mais um carro do estacionamento principal e este estiver vazio. Caso seja mandado inserir mais um carro e o estacionamento principal estiver cheio, é preciso imprimir a mensagem “overflow”.

Entrada

A entrada é formada por uma sequência de pares de entrada. Cada linha possui dois valores, um caractere indicando o comando e um valor inteiro. O par 0 0 indica finalização do programa. O par i x indica inserir o carro de id x no estacionamento, onde x é um número inteiro. O par r x, indica remover o carro de id x, onde x é um número inteiro indicando o identificador do suposto carro estacionado (veja, que o cliente não tem certeza se deixou o carro no estacionamento do Sr. EDCarlos, é uma cidade confusa).

Saída

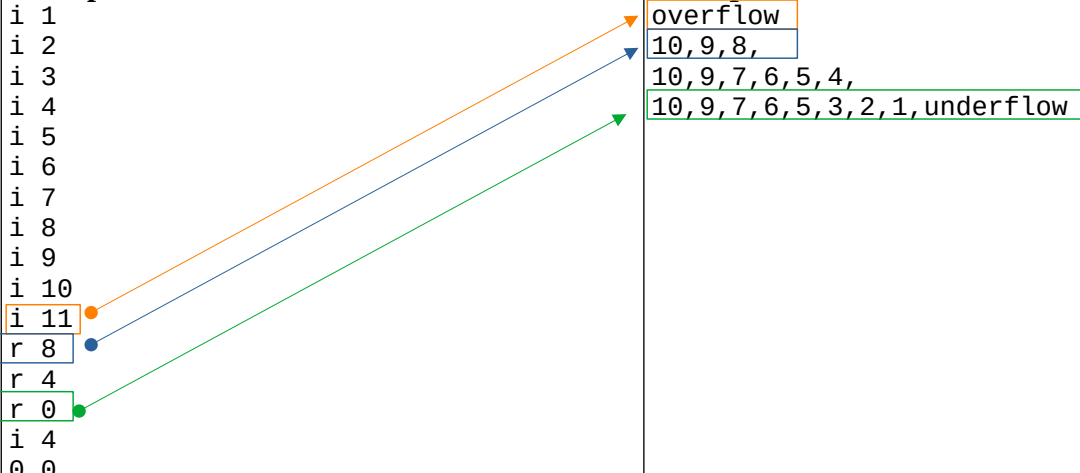
A cada linha, contendo o par de informações que representa uma instrução, deve-se imprimir em apenas duas ocasiões. Caso o par de entrada seja uma instrução de inserir (i), se o estacionamento estiver cheio deve-se imprimir a mensagem “overflow”, sem as aspas e com um ‘\n’ no final. Caso a instrução seja de remoção ‘r’, a cada movimento do carro y do estacionamento principal para o auxiliar, deve ser impresso o id do carro y movido seguido de vírgula; Caso todos os carros sejam movidos do estacionamento principal, e o id procurado para remoção não seja encontrado, deve-se emitir a mensagem “underflow” seguido de um ‘\n’.

Exemplos

Exemplo de entrada 1: i 1 i 2 i 3 i 4 i 5 i 6 i 7 i 8 i 9 i 10 i 11 r 8 r 4 r 0 i 4 0 0	Exemplo de saída 1: overflow 10, 9, 8, 10, 9, 7, 6, 5, 4, 10, 9, 7, 6, 5, 3, 2, 1, underflow
--	---

Explicação

Exemplo de entrada 1: i 1 i 2 i 3 i 4 i 5 i 6 i 7 i 8 i 9 i 10 i 11 r 8 r 4 r 0 i 4 0 0	Exemplo de saída 1: overflow 10, 9, 8, 10, 9, 7, 6, 5, 4, 10, 9, 7, 6, 5, 3, 2, 1, underflow
--	---



O quadrado laranja, indica que não se pode adicionar mais um carro, pois a capacidade máxima do estacionamento é 10, esta forma emitui-se a mensagem “overflow”. Os itens em azul (instrução r 8), indica a solicitação de remoção do carro de id=8. Neste caso, gerou-se o movimento dos carros 10, 9, e 8. Os carros 10 e 9 foram para o estacionamento auxiliar para retornar depois, enquanto que o 8 foi removido para o cliente. O quadrado verde, identificando a instrução r 0, fez com que todos os carros do estacionamento principal fossem movidos, porém encerrou com uma mensagem “underflow”, indicando que o carro procurado não estava no estacionamento. Os carros retornam para o estacionamento principal.