

Hashing: Sondagem Linear

Limite de tempo: 1s
Limite de memória: 256MB

Autor: Daniel Saad Nogueira Nunes

No método da sondagem linear, não se utiliza encadeamento, as colisões de uma função de hashing são tratadas da seguinte forma: se k é a chave, m é o tamanho da tabela e $h(k)$ é a posição da tabela dada pela função de hashing, o método procura a próxima posição livre na tabela de hash através da equação $h(k) + 1 \bmod m$. Caso essa posição também não esteja vazia, o método continua verificando as posições $h(k) + 2 \bmod m, \dots, h(k) + m - 1 \bmod m$, até encontrar o primeiro espaço vazio, ou concluir que a tabela está cheia.

Crie um programa que determina quantas sondagens pelo método da sondagem linear são necessárias para inserir ou buscar uma chave em uma tabela de hash, ou para concluir que a tabela está cheia. Considere que a função de hash é da forma:

$$h(k) = k \bmod m$$

Entrada

A primeira linha da entrada possui dois inteiros q e m indicando, respectivamente, o número de operações a serem realizadas, e o tamanho da tabela de hash.

As próximas q linhas contém, cada, dois inteiros, o e k , que indicam, respectivamente, a operação a ser realizada e a chave a ser inserida ou buscada:

- Se $o = 1$, a operação é uma inserção da chave k na tabela de hash.
- Se $o = 2$, a operação é uma busca da chave k na tabela de hash.

Restrições

- $1 \leq q \leq 10^3$;
- $1 \leq m \leq 10^3$;
- $o \in \{1, 2\}$;
- $0 \leq k \leq 10^9$;
- Caso a operação seja de inserção, é garantido que a chave k não está presente na tabela de hash.

Saída

Para cada operação, o programa deve imprimir uma linha com dois inteiros, r e n , que representam, respectivamente, o resultado da operação e o número de sondagens realizadas. O resultado r deve ser:

- -1 se não for possível inserir ou encontrar a chave;
- x a posição da chave k na tabela de hash, após uma inserção ou busca bem sucedida.

Exemplo

Entrada	Saída
13 5	2 1
1 2	3 2
1 12	4 3
1 22	0 1
1 10	1 2
1 30	-1 5
2 3	2 1
2 2	3 2
2 12	-1 5
2 21	4 3
2 22	0 1
2 10	-1 5
2 7	1 2
2 30	