

Uma *hash* completa

Faça um programa que manipula uma tabela de espalhamento respeitando as operações de inserção, remoção e busca.

- A função de espalhamento será $h(k) = \lfloor m \times (kA - \lfloor kA \rfloor) \rfloor$ $h(\diamond) = \lfloor \diamond \times (\diamond - \lfloor \diamond \rfloor) \rfloor$ (método da multiplicação);
- O tamanho da tabela, m e o valor da constante A serão dados no caso de teste;
- As chaves k_1, k_2, \dots, k_N $1, 2, \dots$ serão valores numéricos não negativos;
- Trate colisões pelo método da sondagem quadrática. A função de sondagem deverá ser $h'(k, i) = (h(k) + i^2) \bmod m$ $h'(\diamond, \diamond) = (h(\diamond) + \diamond^2) \bmod \diamond$.

Seu programa deverá ler uma série de operações e, após cada operação, imprimir uma mensagem indicando o resultado.

Entrada

A entrada começa com os valores de m e A , tais que $0 < m < 10^9$ $0 < A < 10^9$ e $0 < A < 10 < A < 1$.

Em seguida aparece uma quantidade arbitrária de linhas na forma $l\#$, sendo que l representa uma letra e $\#$ é um número. Não haverá espaço entre esses símbolos (veja mais abaixo dicas para a leitura).

Para cada linha, a letra é um código de uma operação e o número é uma chave:

- $i\#$: insere uma chave na tabela;
- $r\#$: remove uma chave da tabela;
- $p\#$: procura uma chave na tabela;
- $x\#$: encerra o programa (o número pode ser ignorado).

Saída

Para cada comando, você deve imprimir uma das seguintes linhas.

- "Chave # inserida no indice #"
- "Chave # nao esta na tabela"
- "Chave # removida"
- "Chave # encontrada no indice #"

Dicas

Em Python, você pode fazer usar a seguinte função para ler cada linha da entrada:

```
def leComando():  
    linha = input()  
    return linha[0], int(linha[1:])
```

Essa função retorna uma tupla. Ela pode ser usada assim:

```
comando, chave = leComando()  
while comando != "x":  
    # processe o comando aqui...  
  
    comando, chave = leComando()
```

Em C, você pode fazer o seguinte laço para ler a entrada:

```
char comando;  
int chave;  
while (scanf(" %c%d", &comando, &chave), comando != 'x') {  
    // processe a chave aqui  
}
```

Observe que existe um espaço antes do `%c` na chamada da função `scanf()` acima. Ele é necessário para que o programa "salte" os espaços em branco antes de encontrar o código do comando.

Exemplos de Entrada e Saída

Entrada	16 0.543 i42 i20 i7 r42 p42 p13 p7 i0 r7 p0 p7 x0
Saída	Chave 42 inserida no indice 12 Chave 20 inserida no indice 13 Chave 7 inserida no indice 0 Chave 42 removida Chave 42 nao esta na tabela Chave 13 nao esta na tabela Chave 7 encontrada no indice 0 Chave 0 inserida no indice 1 Chave 7 removida Chave 0 encontrada no indice 1

	Chave 7 nao esta na tabela
Entrada	64 0.6180339887 i42 i220 i830 p220 r42 r220 i220 p830 i42 p830 x0
Saída	Chave 42 inserida no indice 61 Chave 220 inserida no indice 62 Chave 830 inserida no indice 1 Chave 220 encontrada no indice 62 Chave 42 removida Chave 220 removida Chave 220 inserida no indice 61 Chave 830 encontrada no indice 1 Chave 42 inserida no indice 62 Chave 830 encontrada no indice 1