Tabelas Hash

Crie as classes TabelaHash e NohHash e um programa principal em Python:
- O construtor da classe TabelaHash deverá receber uma função que deverá ser utilizada posteriormente para retornar a posição da tabela;
- A classe NohHash deve ter os atributos: idNoh e uma lista listaDados, sendo que o idNoh será utilizado pela função de hash para o cálculo da posição na tabela hash, assim como para a pesquisa de elementos;
- A tabela hash deverá armazenar uma lista de nós para cada entrada da tabela. Portanto, as colisões que porventura ocorram deverão ser pesquisadas na respectiva lista de acordo com o resultado da função de hash;
- O uso da classe TabelaHash deverá ser precedido pela criação de uma função de hash que deverá ser repassada ao seu método __init__.
- Criar o método Insere(noh), sendo noh um objeto da classe NohHash. Esta função deverá inserir o nó na posição correta de acordo com a função de hash. Use uma lista de nós para tratar as colisões;
- Criar o método Atualiza(noh), sendo noh um objeto da classe NohHash. Este método deverá substituir o nó armazenado anteriormente pelo noh fornecido;
- Criar o método Exclui(noh), sendo noh um objeto da classe NohHash. Este método deverá apagar o nó fornecido através do uso do seu identificador (idNoh);
- Criar o método Pesquisa(noh), sendo noh um objeto da classe NohHash. Este método deverá retornar a lista de dados (listaDados) armazenada dentro do nó;
- A classe TabelaHash deverá armazenar todos os dados em uma única lista em python. Os nós serão armazenados nesta lista pelo seu programa. Lembre-se que podem ocorrer colisões de elementos na mesma posição da tabela, portanto você deve armazenar uma lista de nós em cada posição (sua tabela hash será uma "lista de listas de nós").

Por fim, use as classes TabelaHash e NohHash em um programa principal para tratar as entradas e saídas conforme segue:

Entrada

O arquivo de entrada consiste de várias duplas ou triplas de valores em uma única linha. Cada tripla contém as informações: OP ID LISTA, onde OP é uma operação dentre "insert" ou "update". Por outro lado, cada dupla contém somente as informações OP ID onde OP é uma operação dentre "delete" ou "query". Em todo caso, ID é o identificador (número inteiro) do elemento (este identificador será utilizado pela função de hash), e nas triplas a LISTA é uma lista de informações entre colchetes a ser armazenada no nó. Os elementos (duplas ou triplas) serão separadas por !!! (três exclamações) na entrada.

Inclua, no seu programa, o arquivo "funcHash.py" que contém a função de hash "funcaoHash". Esta função deverá ser utilizada no seu programa principal ao criar o objeto da classe TabelaHash (será fornecida como parâmetro ao método __init__).

Saída

A saída deve conter apenas o resultado da opção query separados por !!!. Portanto, as operações insert, update e delete não devem gerar saídas, pois somente deverão alterar o estado interno da tabela hash. Já a opção query deverá, para cada ID fornecido, imprimir a respectiva lista armazenada junto a este ID no nó. Todas as saídas das query's serão separadas por três exclamações.

Exemplos

Entrada1: `insert 3 ['abracadabra', 30]!!!insert 10 ['pe de cabra', 100]!!!query 3`	Saída1: `['abracadabra', 30]`
Entrada2: `insert 3 ['abracadabra', 30]!!!insert 10 ['pe de cabra', 100]!!!query 3!!!query 10`	Saída2: `['abracadabra', 30]!!![pe de cabra', 100]`