

**A interpretação da prova faz parte da avaliação, portanto leia com atenção.**

Sua tarefa nesta prova é fazer um programa que deverá ser capaz de realizar inserções e buscas de elementos em tabela hash. Cada elemento contém: um texto com até 128 caracteres e um número inteiro. Cada elemento é portanto composto por esses dois valores, sendo o texto considerado a chave.

O programa recebe via entrada, nessa ordem:

- o número de elementos a serem inseridos  $n$
- $n$  elementos, sendo primeiro um texto de até 128 caracteres e a seguir um número inteiro
- o número  $b$  de buscas a serem realizadas na tabela hash
- $b$  buscas representadas por uma chave em formato texto de até 128 caracteres

A tabela hash deve ser criada com tamanho  $m = \lfloor 1,3 \cdot n \rfloor$  (arredondando para baixo), ou seja com 30% a mais de espaço do que o total de elementos a serem inseridos  $n$ .

Como a chave é uma string, a função hash deve considerar os caracteres para retornar a posição de inserção/busca. Utilize a seguinte função hash  $h(.)$  que considera os caracteres de 4 em 4:

$$h(s) = \left( \sum_{i=0}^{l-1} s[4i] \cdot 13^i \right) \bmod m,$$

sendo  $l$  o tamanho da string.

Ou seja, você deverá, a cada 4 caracteres, multiplicá-lo por 13 elevado à potência de  $i$ . Note que  $i$  é incrementado de um em um.

Exemplo, para a string "abcdefghi", e considerando que o caracter 'a' equivale ao 97, 'b' equivale ao 98 e assim por diante:

$$h("abcdefghi") = ((a' \cdot 13^0) + (e' \cdot 13^1) + (i' \cdot 13^2) + c) \% m,$$

$$h("abcdefghi") = ((97 \cdot 13^0) + (101 \cdot 13^1) + (105 \cdot 13^2) + c) \% m,$$

sendo que  $c$  representa o número de colisões obtidas até o momento para a chave atual.

**Exemplo de entrada** com  $n = 5$  elementos e  $b = 2$  buscas. Veja que  $m = \lfloor 1,3 \cdot 5 \rfloor = \lfloor 6,5 \rfloor = 6$

```
5
abc
1
efg
2
hij
3
lkm
4
nop
5
2
nop
aee
```

A saída deve ser da seguinte forma: após a inclusão de todos os  $n$  elementos, imprimir na tela cada elemento da tabela, em ordem, uma em cada linha, considerando apenas as posições ocupadas (pulando as livres), no formato: `%d %s-%d\n`, sendo a posição na tabela hash, espaço, o texto, um hífen seguido do número inteiro associado.

Após a leitura do valor  $b$ , são entradas strings para busca. Para cada string realizar a busca. Caso a busca seja com sucesso exibir o elemento no formato `%s-%d\n`. Caso a busca seja sem sucesso exibir apenas a string e o valor "não encontrado"(NE), no formato `%s-NE\n`

**Exemplo de saída** conforme entrada anterior

```
0 lkm-4
1 abc-1
2 hij-3
3 nop-5
5 efg-2

nop-5
aee-NE
```

Note que o elemento 4 não é exibido por estar livre. Note ainda que após a lista dos elementos há um espaço em branco, seguido do resultado das duas buscas.

Após a implementação, responda as questões no Moodle.