

Uma hash para strings

Faça um programa que lê strings e as insere em uma tabela de espalhamento. Ao final, imprima o conteúdo da tabela.

As chaves serão sequências de caracteres $k = x_0x_1x_2\dots x_n$ tais que cada x_i é uma letra minúscula ou o símbolo hífen e $n < 30$. Ou seja, serão palavras com no máximo 30 caracteres contendo apenas letras e hífens. Por exemplo, `codebench` e `a-b-c` são chaves válidas, mas `CodeBench`, `a b c` e `order-66` não aparecerão na entrada.

Para uma chave k de tamanho n e uma tabela com tamanho $m \in \mathbb{N}^+$, a sua função de espalhamento deve ser

$$h(k) \equiv (\sum_{i=0}^n f(x_i)) \pmod{m} \quad h(i, j) \equiv (\sum_{i=0}^n f(x_i) \cdot (j^i)) \pmod{m},$$

sendo que $f(x_i)$ é uma função que atribui um valor numérico para cada caracter x_i . Essa função deve ser tal que $f(-) = 0$, $f(a) = 1$, $f(b) = 2$, $f(c) = 3$ e assim por diante até $f(z) = 26$.

Para a sondagem linear, em caso de colisão, use a seguinte relação:

$$h'(k, i) \equiv (h(k) + i) \pmod{m} \quad h'(i, j) \equiv (h(i) + j) \pmod{m}.$$

Entrada

A entrada começa com um valor inteiro $0 < m < 8000$ indicando o tamanho que sua tabela de espalhamento deve ter, seguida por um inteiro $0 < n \leq m$ indicando o número de valores que devem ser inseridos na tabela.

Após a configuração da tabela, aparecerão as chaves inteiros $x_1, x_2, x_3, \dots, x_n$. As chaves serão palavras contendo apenas letras minúsculas e o caracter hífen, com no máximo 30 caracteres.

Saída

Imprima apenas as posições não vazias da tabela usando a seguinte formatação:

- Em C/C++: `"%d: %s\n", índice, tabela[índice]`
- Em Python: `"%d: %s" % (índice, tabela[índice])`

Dicas

- Você pode "percorrer" uma string com `for letra in palavra` em Python ou com `for (i = 0; palavra[i]; i++)` em C/C++.

- Em Python, a expressão `ord(letra)` "transforma" um caracter em um valor numérico. Em C/C++, um caracter pode ser usado diretamente como um valor numérico.
- Em Python e em C, as letras do alfabeto romano são sequenciais. Você pode usar a expressão `ord(letra) - ord('a')` como parte da sua função $h(k,i)h(\diamond, \diamond)$; em C/C++ a expressão equivalente é simplesmente `letra - 'a'`.

Exemplos de Entrada e Saída

Entrada	7 5 a ab d a-d aba-bab
Saída	1: a 2: aba-bab 3: ab 4: d 5: a-d
Entrada	11 5 alfa codebench osga hash code-bench
Saída	3: hash 4: codebench 5: code-bench 9: alfa 10: osga