

Problema 5 - Ordenação Linear aplicada a *strings*

Tempo limite: 0,5s (C)

O *Counting sort* é um algoritmo de ordenação adequado para ordenar N inteiros de valores 0 a k e $k=O(N)$. Porém, este método apresenta restrição para ser aplicado na classificação de um conjunto de *strings*, pois para isso seria preciso realizar conversão de base numérica e k se tornaria muito grande para *strings* compridas.

O *Radix sort*, por outro lado, pode ser aplicado para ordenação de *strings* utilizando o *Counting sort* para ordenar a entrada em relação a apenas uma posição das *strings* por vez, sendo que cada caractere seria considerado como um “dígito” do algoritmo. Considerando que *strings* podem ter comprimentos diferentes, um caractere adicional (espaço em branco, ou o número 32 da tabela ASCII em decimal, por exemplo) pode ser utilizado para preencher posições vazias de *strings* mais curtas, para que todas *strings* tenham o mesmo comprimento d (tamanho da maior *string*).

COUNTING-SORT(A, B, k)

```
1  let  $C[0..k]$  be a new array
2  for  $i = 0$  to  $k$ 
3       $C[i] = 0$ 
4  for  $j = 1$  to  $A.length$ 
5       $C[A[j]] = C[A[j]] + 1$ 
7  for  $i = 1$  to  $k$ 
8       $C[i] = C[i] + C[i - 1]$ 
10 for  $j = A.length$  downto 1
11      $B[C[A[j]]] = A[j]$ 
12      $C[A[j]] = C[A[j]] - 1$ 
```

RADIX-SORT(A, d)

```
1  for  $i = 1$  to  $d$ 
2      use a stable sort to sort array  $A$  on digit  $i$ 
```

Deve-se implementar um programa que leia da entrada um conjunto de nomes com até 15 caracteres cada (cada caractere é uma letra dentro dos intervalos $[A, Z]$ e $[a, z]$) e que ordene-os em ordem alfabética. Você deverá converter todas as letras maiúsculas para letras minúsculas antes de realizar a ordenação.

A ordenação deve ser realizada utilizando o algoritmo de ordenação *Radix sort* e o *Counting sort* para ordenar cada posição das *strings*. O número de dígitos d deve ser o comprimento da maior *string* presente na entrada de cada caso de teste. Para cada “dígito”, você deve imprimir os valores em cada posição do vetor auxiliar C após a execução da linha 8 do algoritmo *Counting sort* apresentado acima. O vetor C deve ser de tamanho $k=27$, sendo a primeira posição destinada ao caractere adicional (vazio) e as posições restantes referentes às 26 letras minúsculas em ordem crescente (a, \dots, z).

Após a ordenação, você deve imprimir uma região indicada da lista ordenada com os nomes em letras minúsculas.

Entrada

A primeira linha da entrada contém o número N ($2 \leq N \leq 100.000$) referente a quantidade de nomes a serem ordenados. As próximas N linhas devem conter os respectivos nomes de acordo com a ordem de entrada, sendo um nome para cada linha.

A linha seguinte indica o que deverá ser impresso na saída, e contém dois números: P ($1 \leq P \leq N$) referente a posição do primeiro nome da lista ordenada a ser impresso, e o número M ($1 \leq M \leq N-P+1$) referente a quantidade de números a partir do P -ésimo nome da lista ordenada.

Saída

Para cada “dígito” i do *Radix sort*, imprima uma linha com os 27 valores do vetor C do *Counting sort*. Nas próximas M linhas, imprima os M nomes presentes a partir da P -ésima posição da lista ordenada, um nome a cada linha sem os caracteres adicionais.

Exemplo de Entrada																													
5																													
Daniel																													
Ana																													
Jose																													
Silvia																													
Joao																													
1 5																													
Exemplo de Saída																													
3 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5 5 5 5 5 5 5																													
3 3 3 3 3 4 4 4 4 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5																													
1 1 1 1 1 2 2 2 2 3 3 3 3 3 3 4 4 4 4 4 4 4 5 5 5 5 5																													
0 2 2 2 2 2 2 2 2 2 2 2 3 3 4 4 4 4 4 5 5 5 5 5 5 5 5																													
0 1 1 1 1 1 1 1 1 2 2 2 2 2 3 5 5 5 5 5 5 5 5 5 5 5 5																													
0 1 1 1 2 2 2 2 2 2 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5																													
ana																													

daniel
joao
jose
silvia

Exemplo de Entrada

10
Zurich
Zu
Wu
Hu
Zukemberg
Wei
Woo
Zoo
Zuu
aaaaa
1 10

Exemplo de Saída

9 9 9 9 9 9 9 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 10 10 10 10 10 10 10 10
9 9 9 9 9 10
8 8 9 9 9 9 9 9 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
7 8 8 9 9 9 9 9 9 9 9 9 10 10 10 10 10 10 10 10 10 10 10 10
7 8 8 8 8 9 9 9 9 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
3 4 4 4 4 4 4 4 4 5 5 6 6 6 6 8 8 8 9 9 9 10 10 10 10 10
0 1 1 1 1 2 2 2 2 2 2 2 2 2 2 4 4 4 4 4 10 10 10 10 10
0 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 5 5 5 10
aaaaa
hu
wei
woo
wu
zoo

zu
zukemberg
zurich
zuu

Exemplo de Entrada

10
USA
brazil
zimbabwe
Burundi
Iraq
Vietnam
Greece
bhutan
Iran
Ukraine
5 5

Exemplo de Saída

9 9 9 9 9 10
6 6 6 6 6 7 7 7 7 8 8 8 8 9 9 9 9 9 9 9 9 10 10 10 10
3 4 5 5 6 7 7 7 7 7 7 7 8 8 10 10 10 10 10 10 10 10 10 10 10
3 5 5 6 6 6 6 6 6 8 8 8 8 8 10 10 10 10 10 10 10 10 10 10
1 2 3 3 3 4 4 4 4 4 4 4 4 5 5 5 6 6 6 8 9 9 9 9 9 10
0 4 4 4 4 6 6 6 6 6 6 6 7 7 7 7 7 9 9 9 10 10 10 10 10
0 0 0 0 0 0 0 0 1 3 3 4 4 4 4 4 4 4 8 9 9 10 10 10 10
0 0 3 3 3 3 3 4 4 6 6 6 6 6 6 6 6 6 6 8 9 9 9 9 10
iran
iraq
ukraine
usa
vietnam