

AED2 2022 (1s) - EXERCÍCIO 7 - ORDENANDO PALAVRAS 2.0

Instruções:

1. E/S: tanto a entrada quanto a saída de dados devem ser "secas", ou seja, não devem apresentar frases explicativas. Siga o modelo fornecido e apenas complete as partes informadas (veja o exemplo abaixo).
2. Identificadores de variáveis: escolha nomes apropriados
3. Documentação: inclua cabeçalho, comentários e indentação no programa.
4. Submeta o programa no sistema judge: <http://judge.unifesp.br/aed2S01A2022/>

Descrição:

O grupo de amigos achou divertido brincar com o jogo "*Brincando com as palavras ordenadas*" e resolveu continuar com a brincadeira. Mas desta vez, eles acharam que seria mais divertido não excluir palavras e também tratá-las como se tivessem o mesmo comprimento (medido em número de caracteres). Por isso, incluíram mais duas regras à brincadeira:

1. se a palavra contiver letras **maiúsculas**, elas devem ser convertidas para **minúsculas**;
2. espaços em branco devem ser utilizados para deixar todas as palavras com o mesmo comprimento da **maior** palavra. Cada palavra tem no **máximo 20** caracteres.

Assim como no outro jogo, cada participante terá um papel no jogo. O jogo começa com um dos participantes sorteando um número inteiro positivo N que representa a quantidade de palavras, depois o segundo participante escreve as N palavras em um papel, e assim por diante. Veja o exemplo abaixo:

Exemplo:

- O primeiro jogador especifica 7 como o número de palavras do conjunto original de entrada.
- O segundo jogador escreve as palavras desse conjunto: "**programar VAMOS palavra eh futebol computador legal**".
- O terceiro jogador tem a função de definir uma sub-lista do vetor ordenado para ser exibida como resultado da saída. Para isso o jogador especifica dois números (P e M), como por exemplo, "3 2", os quais indicam a sub-lista que deverá ser exibida na saída. O primeiro valor P ($1 \leq P \leq N$) refere-se a posição/índice do primeiro nome da **lista ordenada** a ser exibido, e M ($1 \leq M \leq N - P + 1$) refere-se a quantidade de números a serem exibidos a partir do P -ésimo nome da mesma **lista ordenada**. No caso do exemplo aqui citado, requer-se que seja exibido como sub-lista de saída, o nome correspondente a posição 3 do vetor ordenado,

seguido do próximo nome correspondente a posição adjacente 4, uma vez que deverão ser exibidos dois valores a partir da posição 3 do vetor ordenado.

- O quarto transforma maiúscula em minúscula.
- O quinto descobre o tamanho da maior palavra e completa todas com espaço em branco para todas terem o tamanho máximo de 20 caracteres.
- O sexto jogador ordena o conjunto resultante e exibe o vetor auxiliar C após a execução da linha 8 do algoritmo *Counting sort* apresentado pelo monitor. Observação: todas palavras devem ser ordenadas.
- o sétimo jogador exibe a saída requisitada, mostrando inicialmente o mesmo vetor de entrada com as palavras convertidas para minúsculo. Depois deve mostrar um número inteiro " d " que corresponde ao tamanho da maior palavra. Posteriormente, exibe as d linhas do vetor C do *Counting sort*. Para finalizar, deve mostrar a sub-lista do vetor ordenado de acordo com o especificado pelo terceiro jogador.

Considere as seguintes condições:

1. Sua solução deve implementar o algoritmo da Figura 1 para ordenar e **nenhum** outro pode ser usado;
2. O número de dígitos d deve ser o comprimento da maior string presente na entrada de cada caso de teste.
3. Para cada "dígito", deve-se imprimir os valores em cada posição do vetor auxiliar C após a execução da linha 8 do algoritmo *Counting sort*. 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, b, ..., z).
4. O código-fonte **deve** ser escrito em C/C++ ou Java;
5. **Toda** memória alocada dinamicamente (C/C++) deve ser desalocada;
6. **Nenhuma** variável global deve ser utilizada;

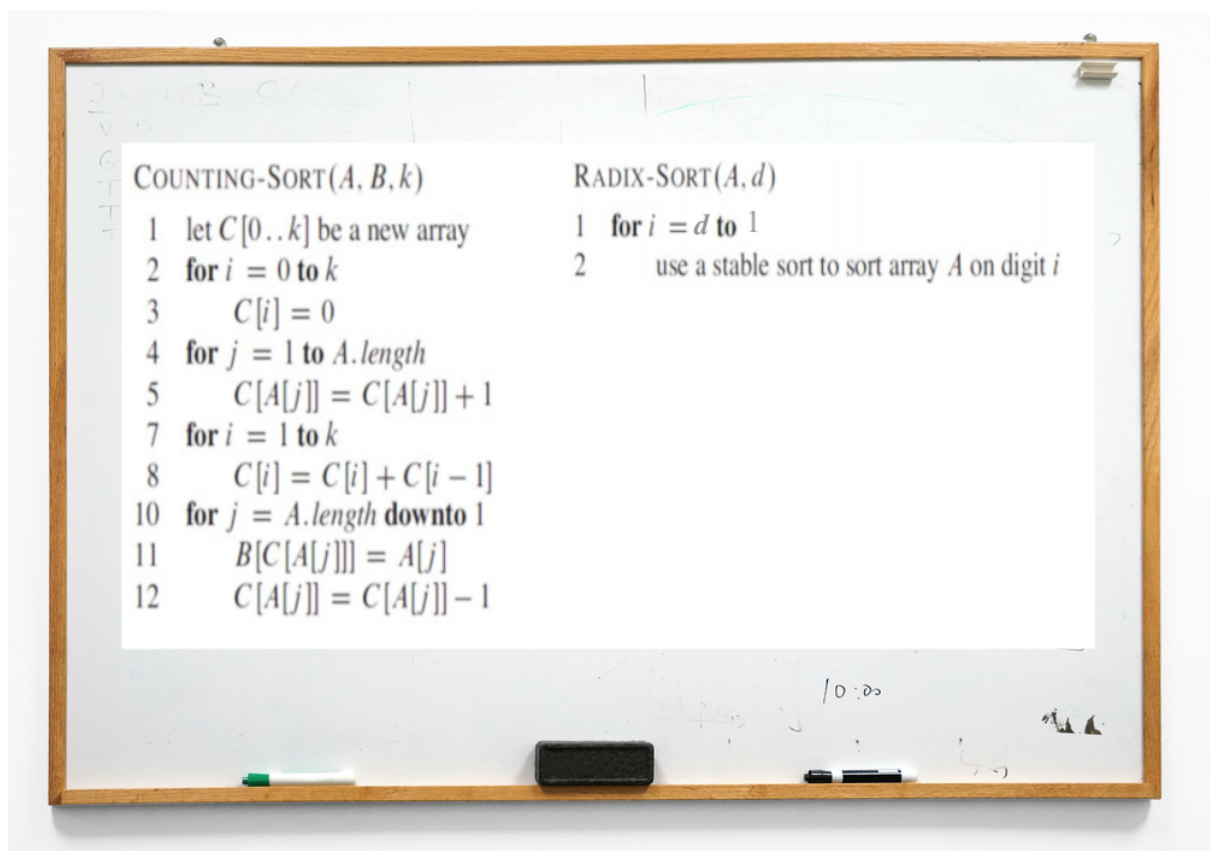


Figura 1: Algoritmo

ENTRADA:

Primeira linha contém a quantidade N do total de palavras do conjunto inicial.

A segunda linha contém as N palavras separadas por um espaço em branco, representando o conjunto inicial de palavras: p_1, p_2, \dots, p_N .

A linha seguinte indica o que deverá ser exibido como sub-lista de 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 serem exibidos a partir do P -ésimo nome da **lista ordenada**.

SAÍDA:

Inicialmente deve-se exibir na saída o mesmo vetor de entrada com as palavras convertidas para caracteres minúsculos. Cada palavra deve ser acrescida do sinal ponto final (".") no fim.

Posteriormente, deve-se exibir um valor " d " correspondente a quantidade de caracteres da maior palavra do vetor de entrada.

Em seguida, para cada "dígito" i do *Radix sort*, imprima uma linha com os 27 valores do vetor C do *Counting sort*. Devem ser exibidas " d " linhas no total.

Finalizando, nas próximas M linhas, imprima as M palavras presentes a partir da P -ésima posição da lista ordenada, uma palavra a cada linha sem os caracteres adicionais (Obs: Nesta sub-lista ordenada as palavras **não tem** o sinal de ponto final exibido no fim).

Exemplos de entrada e saída

Importante: o sinal de ponto final (".") é apenas na impressão do vetor de entrada convertido para minúsculo, e não deve ser guardado com a palavra.

[illegible]

Tabela 1: Exemplos de entrada e saída 01

[illegible]

Exemplos de entrada
10 banana amora jabuticaba acerola cabeluda jambo abacaxi laranja limao abacate 1 4
Exemplos de saída
banana. amora. jabuticaba. acerola. cabeluda. jambo. abacaxi. laranja. limao. abacate. 10 9 10 9 9 10 8 10 4 6 6 7 8 9 9 9 9 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 3 4 4 4 4 4 4 4 5 6 6 7 7 7 7 7 7 7 8 9 9 9 10 10 10 0 3 3 3 3 3 3 3 3 3 3 4 4 6 9 9 9 9 10 10 10 10 10 10 10 0 3 4 6 6 7 7 7 7 7 7 7 7 7 7 7 9 9 9 10 10 10 10 10 10 0 2 4 4 4 5 5 5 5 5 5 5 7 8 9 9 9 10 10 10 10 10 10 10 10 0 5 7 8 8 8 8 8 8 9 9 9 9 10 10 10 10 10 10 10 10 10 10 10 10 0 4 5 6 6 6 6 6 6 6 8 8 10 10 10 10 10 10 10 10 10 10 10 10 10 10 abacate abacaxi acerola amora

Tabela 3: Exemplos de entrada e saída 03