AED2 2021 - TAREFA 7 - BRINCANDO COM AS PALAVRAS ORDENADAS V-2.0

Entrega: 26/05/2021 até 23:59:59

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 utilizando acesso remoto via VPN: `http://judge.sjc.unifesp.br/aed2`, ou através de conexão direta: `http://kp.unifesp.br:9001/aed2/login`.
5. O código-fonte pode ser escrito em C, C++ ou Java.
6. **O código-fonte DEVE implementar uma solução usando o algoritmo Radix-Sort e Counting sort.**

Descrição:

O Prof. de de AEDII programação elogiou seus alunos pela brincadeira proposta *"Brincando com as palavras ordenadas"*. Por isso, a turma resolveu fazer um novo jogo utilizando um novo algoritmo de ordenação: desta vez, a regra é utilizar o *RadixSort* e *Counting sort*. Como existem diversas forma de ordenar, o monitor da turma achou melhor escrever o algoritmo no quadro (veja a Figura 1).

A turma achou que seria mais divertido não excluir palavras e também tratá-las como se tivessem o mesmo comprimento (medido em número de caracteres). Assim, incluiu mais duas regras: 1) se a palavra for **maiúscula**, ela deve ser convertida para **minúscula** e 2) espaços em branco devem ser utilizados para deixar todas as palavras com o mesmo comprimento da **maior** palavra. 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 ($N < 1000$) palavras em um papel.

O jogo tem algumas restrições:

1. A quantidade máxima de palavras é menor que 1000.
2. Cada palavra tem no máximo 20 caracteres.
3. É requisito **obrigatório** usar o algoritmo que o monitor escreveu no quadro para ordenar e nenhum outro pode ser usado, senão será zerado não só no jogo mas também na nota!(òló)

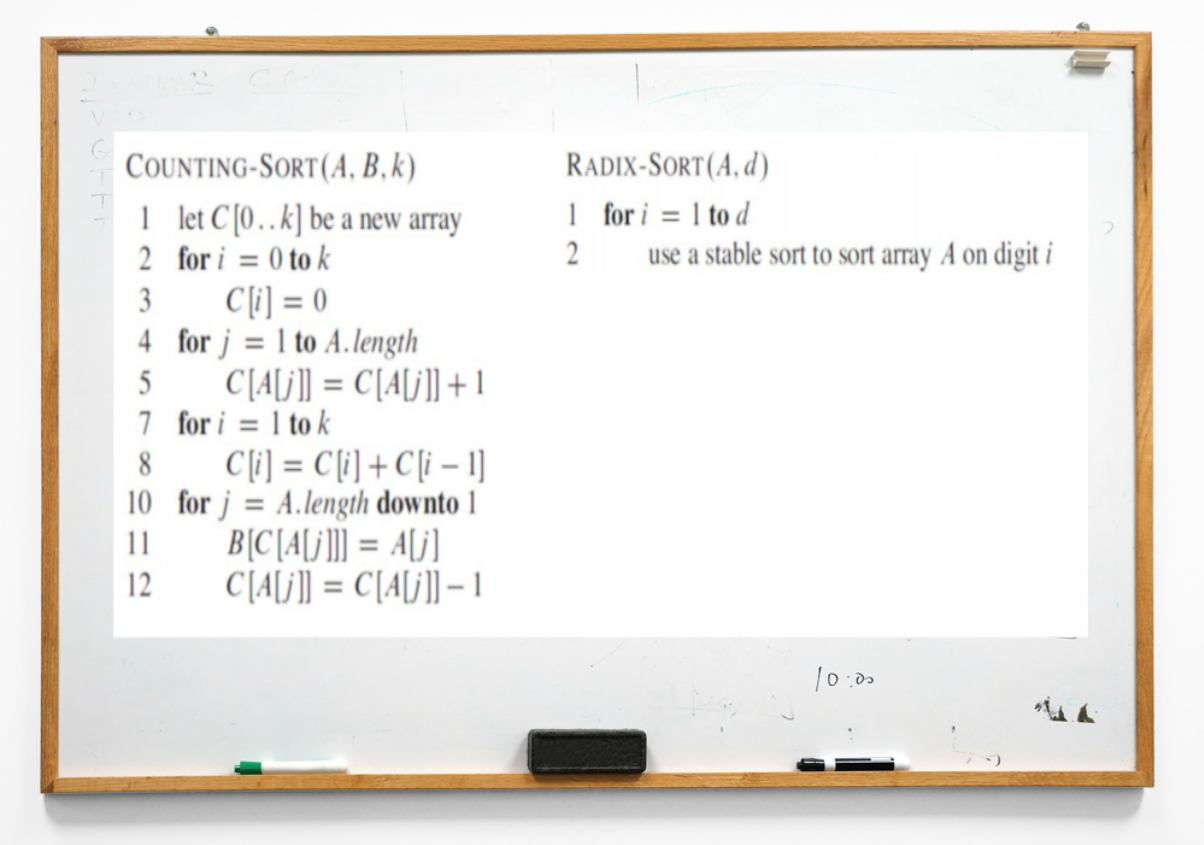

Figura 1: Algoritmo

4. O número de dígitos d deve ser o comprimento da maior string presente na entrada de cada caso de teste.

5. 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*. 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,$ldots$, z).

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, requere-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 sexta 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-se 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. **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.

E N T R A D A:

Primeira linha contem 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: $p_1, p_2, ..., p_N$.

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**.

S A Í D A:

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:

Exemplos de entrada
7 programar VAMOS palavra eh futebol computador legal 3 2
Exemplos de saída
programar. vamos. palavra. eh. futebol. computador. legal. 10 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 6 6 6 7 7 7 7 7 7 7 7 5 6 6 6 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 3 5 5 5 5 5 5 5 5 5 5 5 6 7 7 7 7 7 7 7 7 7 7 7 7 7 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 6 6 7 7 7 7 7 7 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 4 5 5 6 7 7 7 7 1 3 3 3 3 4 4 5 5 5 5 5 5 5 5 6 7 7 7 7 7 7 7 7 7 7 1 1 1 1 1 1 1 2 2 2 2 2 3 5 5 6 6 6 6 6 7 7 7 7 7 7 0 2 2 2 2 3 3 3 4 4 4 4 4 4 4 5 5 5 6 6 6 7 7 7 7 7 0 0 0 1 1 2 3 3 3 3 3 3 4 4 4 4 6 6 6 6 6 6 7 7 7 7 futebol legal

Tabela 1: Exemplos de entrada e saída 01

Exemplos de entrada
6 Vaca Calopsita gaTO zEbra COBRA COruJA 1 6
Exemplos de saída
vaca. calopsita. gato. zebra. cobra. coruja. 9 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 6 6 6 6 6 6 6 5 5 5 5 5 5 5 5 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 4 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 6 6 6 6 6 6 6 2 4 4 4 4 4 4 4 4 4 5 5 5 5 5 6 6 6 6 6 6 6 6 6 6 0 1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 3 5 5 5 6 6 6 6 6 0 0 2 3 3 3 3 3 3 3 3 4 4 4 4 4 4 5 5 6 6 6 6 6 6 0 3 3 3 3 4 4 4 4 4 4 4 4 4 4 6 6 6 6 6 6 6 6 6 6 0 0 0 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 6 calopsita cobra coruja gato vaca zebra

Tabela 2: Exemplos de entrada e saída 02

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