

# AED2 2023 - EXERCÍCIO 5 - ORDENANDO PALAVRAS

---

## 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:

Num típico sábado à noite, um grupo de amigos resolveu praticar a implementação de algoritmos de ordenação. Para tornar a brincadeira mais interessante, eles decidiram ordenar palavras usando uma chave especial, que altera a ordem das letras do alfabeto. A chave contém todas as letras necessárias para ordenar as palavras, em ordem crescente de valor. O objetivo do teste é ordenar, usando o algoritmo *heapsort*, uma sequência de  $N$  palavras, utilizando a ordem lexicográfica apresentada na chave.

Por definição, a **chave deve conter todos os caracteres** presentes nas palavras fornecidas. Caso seja encontrada uma palavra que não atende a essas condições - ou seja, que contenha um caractere não presente na chave - o programa deve ser encerrado com a mensagem *A palavra [xxx] eh invalida*. Basta imprimir a primeira palavra que contenha um caractere inválido.

Considere as seguintes condições:

1. Sua solução deve implementar *HeapSort*;
2. A complexidade do algoritmo deve se manter em  $O(n \log n)$ ;
3. O código-fonte **deve** ser escrito em C/C++ ou Java;
4. **Toda** memória alocada dinamicamente (C/C++) deve ser desalocada;
5. **Nenhuma** variável global deve ser utilizada;
6. É seguro assumir que nenhuma palavra terá mais que 32 caracteres.

Solução que violem essas condições não serão aceitas.

**Exemplo:**

- São especificadas 2 palavras e uma chave de 6 caracteres.
- A chave fornecida é *pzmora*.
- A entrada contém as palavras *amor* e *paz*.
- Como a letra *p* aparece na chave antes da letra *a*, as palavras são ordenadas como *paz amor*.

**ENTRADA:**

1. A primeira linha contém 2 números inteiros, *N* e *S*, sendo *N* o número de palavras a serem ordenadas e *S* o tamanho da chave.
2. A segunda linha contém a chave de tamanho *S*, com todas as letras necessárias para a ordenação das palavras.
3. A terceira linha contém *N* palavras, separadas por espaços.

**SAÍDA:**

Caso alguma palavra do conjunto de entrada de *N* palavras não esteja de acordo com a regra, a **primeira** palavra fora do padrão deve ser impressa conforme os exemplos exibidos a seguir. Caso todas as palavras do conjunto de entrada estejam de acordo com as regras, a saída deve ser uma única linha contendo as palavras ordenadas de acordo com a ordem lexicográfica apresentada na chave, separadas por espaço. Note que **não deve haver espaço após a última palavra**.

**Exemplos de entrada e saída:**

Exemplos de entrada	Exemplos de saída
5 26 avdbgtcmuhqkrpefzwyoxnjsli interessantes de ordenacao sao algoritmos	algoritmos de ordenacao sao interessantes
6 13 lanmcreogjitb banana maca laranja limao acerola morango	laranja limao acerola maca morango banana
7 26 tdhvgcuoiqrpeafzwyxnjslmb the dear hunter é uma banda genial	A palavra é eh invalida

Tabela 1: Exemplos de entrada e saída