



2ª AVALIAÇÃO

Instruções para a avaliação

Data: 21/10/2020

Horário: 12:30 às 15:30

- Leia a avaliação com atenção.
- Essa avaliação abrange especificamente os conteúdos de funções, vetores, strings e matrizes, além das estruturas sequencial, de decisão e de repetição.
- A avaliação **DEVE** ser resolvida de forma individual, **SEM CONSULTA A QUALQUER TIPO DE MATERIAL**.
- Nenhum arquivo, exceto os da avaliação, deve ser aberto e/ou manipulado durante o período da avaliação.
- A interpretação das questões faz parte da avaliação.
- Os arquivos devem ser nomeados na forma: **NomeDoAlunoExercicioX.c**, sendo X o número da questão.
- Ao final da avaliação, somente o código fonte (.c) de cada questão deverá ser enviado para a tarefa **SEGUNDA AVALIAÇÃO**, que está no tópico **7 - Avaliações** no Moodle.
- É de responsabilidade do aluno a gravação correta dos arquivos da avaliação.
- O tempo de avaliação é das 12h30min às 15h30min.
- A avaliação vale 10 pontos e tem peso 6,0 na média final.
- **As respostas das questões serão submetidas a um software detector de similaridade para verificar a autenticidade do código desenvolvido pelo aluno.**
- Boa Sorte!

Questão 1 (3,3 pontos) - Ler uma string e contar quantos espaços sucessivos ela contém. Armazenar esses valores em um vetor. Utilizando função, mostrar os elementos desse vetor. Identifique se nesse vetor há três valores em sequência que estão em ordem crescente. Se há, mostre-os. Em seguida, usando função, ordene o vetor e mostre-o novamente. Conte a quantidade de cada valor que há no vetor. Por exemplo: String informada (considere no exemplo que os traços representam os espaços):

-----Engenharia-de---Computacao--da----Universidade--Tecnologica---do----Parana-.

Vetor mostrado:

5 1 3 2 4 2 3 4 1

Sequência de três números em ordem crescente:

2 3 4

Vetor ordenado

1 1 2 2 3 3 4 4 5

Valor Quantidade de ocorrências

1 2

2 2

3 2

4 2

5 1

Questão 2 (3,3 pontos) - Ler uma string. Copiar o valor decimal do código ASCII de cada elemento da string para um vetor e mostrar o vetor. Ordenar o vetor e mostrar o vetor ordenado. Percorrer o vetor ordenado contando, sem repetição, os números pares e os números ímpares e mostrar a soma total desses valores. Criar um vetor *pares* e um vetor *impares* e copiar, sem repetição, os elementos pares e os ímpares do vetor ordenado para os vetores *pares* e *impares*, respectivamente, e mostrar os dois vetores. Criar duas matrizes: *mPares* e *mImpares*, sendo que *mPares* deverá conter na primeira coluna todos os elementos pares do vetor principal e na segunda

coluna a ocorrência de cada elemento par no vetor principal e a *mlmpares* deverá conter na primeira coluna todos os elementos ímpares do vetor principal e na segunda coluna a ocorrência de cada elemento ímpar no vetor principal. Mostrar as duas matrizes.

Exemplo:

```
Informe uma string: engenharia

== VETOR ==
101 110 103 101 110 104 97 114 105 97

== VETOR ORDENADO ==
97 97 101 101 103 104 105 110 110 114

Numero de pares (sem repeticao): 3
Numero de impares (sem repeticao): 4

== VETOR PAR ==
104      110      114

== VETOR IMPAR ==
97      101      103      105

== MATRIZ DE PARES ==
PARES  OCORRENCIAS
104     1
110     2
114     1

== MATRIZ DE IMPARES ==
IMPARES OCORRENCIAS
97      2
101     2
103     1
105     1
```

Questão 3 (3,4 pontos) – Criar funções para:

1) Transformar os caracteres alfabéticos de uma string para letras minúsculas. Exemplo:

String: Eu Gosto De Programar

Apresentar: eu gosto de programar

2) Mostrar uma string com apenas 1 espaço entre as palavras e sem nenhum espaço no início e no final da string. Exemplo (considere no exemplo que os traços representam os espaços):

String: ---eu--gosto--de---programar----

Apresentar: eu-gosto-de-programar

Fazer um programa que leia uma string e, usando função, mostre a string em letras minúsculas. Na sequência, também usando função, retirar os espaços do início e do final de uma string e deixar apenas 1 espaço entre as palavras, imprimir a string e calcular e mostrar o percentual de consoantes da string. Ainda, substituir cada vogal lida por N vogais iguais, onde N representa o número de vogais lidas até então, gerando uma nova string. Para os demais caracteres, o programa deverá imprimir apenas o próprio caractere.

Exemplo:

```
Informe um texto:   Eu  Gosto  De  Programar

==== TEXTO EM LETRAS MINUSCULAS ====
  eu  gosto  de  programar

==== TEXTO SEM ESPACOS SUCESSIVOS ====
eu gosto de programar
Percentual de consoantes: 47.62%

==== TEXTO COM VOGAIS REPLICADAS ====
euu gooostoooo deeeee proooooograaaaaaamaaaaaaaar

Deseja executar novamente (S ou N)?
```