



Lista 5 - Estruturas Homogêneas

Vetores de caracteres e de números

Exercícios para sala de aula

Lembretes:

- ✓ Uma forma de percorrer uma string é:

```
while(string[i] != '\0')
```
- ✓ Vetores precisam ter tamanho definido quando declarados.
- ✓ Cuidado para não ultrapassar o tamanho do vetor, ou seja, percorrer índices (ler) ou armazenar valores além do tamanho definido para o vetor.

//Vetores de caracteres

1) Ler uma string e copiá-la para outra string sem os possíveis espaços em branco no início e no final. Também devem ser eliminados dois ou mais espaços sucessivos entre as palavras.

Exemplo:

Informe um texto: Universidade Tecnológica Federal do Paraná

Texto sem espaços: UniversidadeTecnologicaFederaldoParana

Deseja repetir o programa (S ou N)?

2) Elaborar um programa que leia uma frase e determine se a mesma representa um palíndromo. Em frases de palíndromos os espaços não são considerados, então é necessário inicialmente retirar os espaços em brancos do texto. Para comparar as strings use a função `int ComparaStrings(char str1[], char str2[])` criada na Lista 4 de exercícios.

Exemplos de frases palíndromos:

Socorram-me, subi no ônibus em Marrocos

Anotaram a data da maratona

A mala nada na lama

A torre da derrota

A cara rajada da jararaca

Exemplo:

Informe uma frase: a cara rajada da jararaca

Inverso da palavra: acararajadadajararaca

Eh palindromo

Deseja repetir o programa (S ou N)? s

Informe uma frase: a cara da jararaca

Inverso da palavra: acararajadaraca

Nao eh palindromo

Deseja repetir o programa (S ou N)? n

3) Um número primo é definido se ele possuir somente dois divisores: o número um e ele próprio. São exemplos de números primos: 2, 3, 5, 101, 367 e 523. Neste problema, você deve ler uma palavra composta somente por letras [a-zA-Z]. Cada letra possui um valor específico, *a* vale 1, *b* vale 2 e assim por diante, até a letra *z* que vale 26. Do mesmo modo *A* vale 27, *B* vale 28, até a letra *Z* que vale 52. Você precisa definir se cada palavra em um conjunto de palavras é prima ou não. Para ela ser prima, a soma dos valores de suas letras deve ser um número primo.

Para resolver este problema, é necessário criar uma função para contar os caracteres de um texto (com o protótipo `int ContarCaracteres(char texto[]);`) e uma função para converter os caracteres para números (com o protótipo `int ConverteCharParaNumero(char ch);`). Além disso, é necessário usar uma função que recebe um número e retorna 0 se esse número for primo ou 1, caso contrário.

Exemplo:

Informe um texto: a UTFPR campus Pato Branco

a = 1 => Nao prima

UTFPR = 211 => Prima

campus = 73 => Prima

Pato = 78 => Nao prima

Branco = 79 => Prima

Deseja repetir o programa (S ou N)? n

4) Ler uma string e contar quantas palavras a mesma possui. Considerar o seguinte contexto:

O usuário pode informar apenas uma palavra, com ou sem espaços no início ou no final da mesma.

O usuário pode informar mais de um espaço entre palavras.

O usuário pode não informar nenhuma palavra (simplesmente pressionar ENTER).

Exemplo:

```
Informe um texto: UTFPR Pato Branco
"UTFPR Pato Branco" possui 3 palavras
```

```
Deseja continuar no programa (S ou N)? n
```

//Vetores numéricos

5) Gerar um vetor com 10 valores inteiros aleatórios entre 1 e 100. Para os elementos de índice par, verificar se é perfeito. Para os elementos de índice ímpar, somar os seus dígitos. Utilizar funções para verificar se um número é perfeito e para somar os seus dígitos. Um número é perfeito quando a soma dos seus divisores (exceto ele próprio) é igual ao próprio número (por exemplo 6 é perfeito porque possui 1, 2 e 3 como divisores que somam 6).

Exemplo:

```
==== VETOR ====
71  39  14  61  73  59  27  28  24  61
```

```
==== PARES ====
14 nao eh perfeito
28 eh perfeito
24 nao eh perfeito
```

```
==== IMPARES ====
71 - Soma dos digitos: 8
39 - Soma dos digitos: 12
61 - Soma dos digitos: 7
73 - Soma dos digitos: 10
59 - Soma dos digitos: 14
27 - Soma dos digitos: 9
61 - Soma dos digitos: 7
```

```
Deseja continuar no programa (S ou N)?
```

6) Utilizar uma função para gerar um vetor com 10 valores entre 5 e 50. Utilizar uma função para verificar se cada elemento do vetor é ou não um número primo. Armazenar em outro vetor somente os números que são primos. Mostrar os dois vetores utilizando função.

Exemplo:

```
==== VETOR ====
 27  19  40   7  24  15  44  31  31   6
27 nao eh primo
19 eh primo
40 nao eh primo
 7 eh primo
24 nao eh primo
15 nao eh primo
44 nao eh primo
31 eh primo
31 eh primo
 6 nao eh primo
```

```
==== PRIMOS ====
 19   7  31  31
```

Deseja continuar no programa (S ou N)?