

### Lista 3 – Vetores unidimensionais (sala de aula)

Assunto:

Vetores de caracteres

**Lembretes:**

**Utilizar**

**while(String[Cont] != '\0')**

**para percorrer uma string**

**Não esquecer de finalizar uma string, se necessário.**

**Vetores precisam ter tamanho definido quando são declarados.**

**Cuidado para não ultrapassar o tamanho do vetor, ou seja, percorrer índices (ler) ou armazenar valores além do tamanho definido para o vetor.**

**Indispensável:**

Listar todos os exercícios como opções de um menu. Usar as funções gerar e mostrar vetor e criar funções no mesmo arquivo que está a main() para “chamar” as funções que representam cada um dos exercícios.

1) Ler um nome. Se o primeiro caractere for 'a' imprimir esse nome. Se não for contar quantos caracteres 'o' o nome possui. Repetir a leitura até que o usuário informe zero (0) como primeiro caractere do nome.

2) Elaborar um programa que aceita e valida uma senha com 6 dígitos numéricos, que deve ser armazenada em uma string. O usuário tem três tentativas para acertar a senha, se ultrapassar esse limite, ele deve ser informado. O usuário também deve ser informado se ele acertou a senha em até três tentativas.

3) Ler uma string, contar e exibir a quantidade de cada uma das vogais existentes nessa string. Armazenar essas quantidades em um vetor. É indispensável armazenar a quantidade à medida que a string é percorrida (algo como VetVogais[X]++ ou VetVogais[X] = VetVogais[X] + 1). Portanto, o vetor que armazena a quantidade de vogais deve ser inicializado com zero antes de ser utilizado. Mostrar os resultados da seguinte forma.

| Vogal | Quantidade |
|-------|------------|
| ===== |            |
| A/a   | 5          |
| E/e   | 0          |
| I/i   | 1          |
| O/o   | 2          |
| U/u   | 0          |

4) Ler um texto com até 50 caracteres. Percorrer esse texto e verificar quais caracteres são vogais. Criar um vetor armazenando o número da vogal correspondente e somente das vogais (1 para a, 2 para e ....). Para simplificar, considerar que serão informados somente caracteres minúsculos.

Por exemplo:

Informado o texto:

Texto para verificacao de vogais

Saída (vetor gerado):

2 4 1 1 2 3 3 1 1 4 4 1 3

5) Ler uma informação no formato texto (string):

a) imprimir os caracteres na posição par;

b) imprimir os caracteres na posição ímpar;

c) mostrar somente o primeiro caractere de cada palavra, separados por dois espaços;

d) mostrar somente o último caractere de cada palavra, separados por dois espaços;

e) copiar a primeira palavra para outra string e mostrá-la com do while, caractere a caractere. É indispensável finalizar a string copiada, ou seja, colocar \0 no final da mesma.

Exemplo para “mostrar somente o primeiro caractere de cada palavra, separados por dois espaços”.

Informado: “Os ótimos alunos da Universidade Tecnológica Federal do Paraná, campus Pato Branco.”

Mostrado:

“O ó a d U T F d P c P B”

6) Ler uma string, calcular e mostrar quantos caracteres a mesma possui, incluindo espaços e outra contagem não incluindo espaços.

7) Ler um texto imprimi-lo de trás pra frente.

Sugestão: contar quantos caracteres a string possui (tamanho). Para contar percorrer a string até encontrar ‘\0’ que indica final da string.

8) Ler uma frase e imprimir a frase na vertical com uma palavra por linha.

9) Ler uma informação no formato texto, armazenar esse texto lido em um vetor (string) até o primeiro caractere de espaço. Mostrar a string armazenada.

Observação: colocar ‘\0’ na nova string após o último caractere útil para indicar final da mesma.

10) Elaborar um programa que leia uma palavra e determine se a mesma representa um palíndromo. Em frases palíndromas os espaços não são considerados.

Exemplos de palavras palíndromos: ovo, natan, sos, arara.

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

O programa verificará apenas palavras, considerando-as não acentuadas. Pense um pouco sobre como seria um programa que verificaria frases com sinais de pontuação e palavras acentuadas que devem ser desconsiderados na verificação.

11) Ler uma string. Ler um caractere. Pesquisar se esse caractere está na string, se estiver mostrar o índice que o mesmo está armazenado, se não estiver retornar -1.

Para conhecimento: Exemplo de manipulação de strings utilizando switch case:

```
/* Contar e exibir a quantidade de vogais existentes em uma cadeia de
caracteres (string). */

#include "stdio.h";
#include "string.h";
int main(void)
{
    char s[256];
    printf("Informe uma cadeia de caracteres:\n");
    gets(s);

    int ctA = 0, ctE = 0, ctI = 0, ctO = 0, ctU = 0;
    for (int i=0; i<strlen(s); i++)
    {
        switch (s[i])
        {
            case 'a':
            case 'A': ctA = ctA + 1;
                     break;

            case 'e':
            case 'E': ctE = ctE + 1;
                     break;

            case 'i':
            case 'I': ctI = ctI + 1;
                     break;

            case 'o':
            case 'O': ctO = ctO + 1;
                     break;

            case 'u':
            case 'U': ctU = ctU + 1;
                     break;
        }
    }
    printf("\nExistem %d vogais A.\n", ctA);
    printf("Existem %d vogais E.\n", ctE);
    printf("Existem %d vogais I.\n", ctI);
    printf("Existem %d vogais O.\n", ctO);
    printf("Existem %d vogais U.\n", ctU);
}
```