

## Lista de exercícios – Vetores e strings

**Aluno: Adriano dos Santos Elizeu**

1) Fazer um programa de "criptografia" (codificação de dados visando a privacidade de acesso as informações), onde dada uma string (vetor de caracteres) este programa codifique os dados através de um processo de substituição de letras. Você pode definir o seu próprio método de criptografia, desde que depois seja possível reverter este processo, ou seja, um código criptografado deve poder ser convertido novamente ao valor inicial.

```
#include <stdio.h>
int main() {
    char texto[80];
    int i;
    printf("Entre com texto: ");
    scanf("%s", texto);

    for( i = 0 ; i < 79 ; i++){
        if ( texto[i] != '\0'){
            texto[i] = texto[i]+1;
        }
    }
    printf("\nTexto Criptografado: %s", texto);
    return 0;
}
```

2) Fazer um segundo programa, complementar a este anterior, que deve ser capaz de descriptografar a string, ou seja, deve pegar uma string codificada e retornar ao texto original.

```
#include <stdio.h>

int main() {
    char texto[80];
    int i;
    printf("Entre com texto: ");
    scanf("%s", texto);

    for( i = 0 ; i < 79 ; i++){
        if ( texto[i] != '\0'){
            texto[i] = texto[i]-1;
        }
    }
    printf("\nTexto descriptografado: %s", texto);
    return 0;
}
```

3) Fazer um programa que leia uma string e converta todos os caracteres desta string para maiúsculo. Depois de convertida a string, exibir o resultado na tela. Dicas: (1) Uma string de caracteres é um vetor do tipo char, terminado pelo caractere '\0'; (2) O código ASCII da letra 'a' (minúsculo) é 97 e o código da letra 'A' (maiúsculo) é 65, o código da letra 'z' é 122 e o código da letra 'Z' é 90; (3) Cuidado para não acabar convertendo aquilo que não é nem texto e nem está em minúsculas!

```
#include <stdio.h>

int main(){
    char frase[80];
    int i;
    printf("Entre com texto: ");
    scanf("%[^\n]", frase);

    for(i = 0; frase[i] != '\0'; i++){
        if(frase[i] >= 97 && frase[i] <= 122){
            frase[i] -= 32;
        }
    }
    printf("\nTexto Convertido: %s\n", frase);
    return 0;
}
```

4) Faça um programa que leia duas strings (de tamanho máximo 40). Após isso, o programa deve concatenar as informações lidas e mostrar o resultado para o usuário. Exemplo: Se a primeira string digitada for "Bom dia, " e a segunda "moçada !", então o resultado deverá ficar: "Bom dia, moçada !".

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main(){

    char palavra1[40]; char palavra2[40];

    printf("Texto 1: ");
    scanf("%[^\n]*c", palavra1);

    printf("Texto 2: ");
    scanf(" %[^\n]*c", palavra2);

    strcat(palavra1, palavra2);
    printf("\n%s\n", palavra1);
}
```

```
return 0;
}
```

5) Escreva uma função para ler 10 elementos inteiros, colocá-los num vetor e retornar um outro vetor contendo todos os números primos do vetor inicial. Na função, imprima os valores dos dois vetores.

```
#include <stdio.h>
#define LIM 10

int num_primos(int v_1[LIM]){
    int v_2[LIM];
    int i; int j; int cont;

    for (i = 0; i < LIM; i++){
        printf("[%i] Digite qualquer Número: ", i+1);
        scanf("%d", &v_1[i]);
    }
    for (i = 0; i < LIM; i++){
        cont = 1;
        for (j = 2; j < v_1[i]; j++){
            if (v_1[i] % j == 0){
                cont = 0;
            }
        }
        if (cont == 1){
            v_2[i] = v_1[i];
        }
    }
    printf("\nNúmeros primos:");
    for (i = 0; i < LIM; i++){
        printf(" %d ", v_2[i]);
    }

    printf("\nNúmeros não primos:");
    for (i = 0; i < LIM; i++){
        printf(" %d ", v_1[i]);
    }
    return 0;
}

int main (){
    int i, n[LIM];
    num_primos(n);
    return 0;
}
```

6) Implemente um programa que receba um nome completo e apresente apenas o último nome e o 1º nome na seguinte forma:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main(){
    char nome[40];
    char meio[40];
    char ultimo[40];

    printf("Digite seu nome: ");
    scanf("%[^ ]%c", nome);
    scanf("%[^ ]%c", meio);
    scanf("%[^\n]%*c", ultimo);

    printf("\nSaída: %s, %s\n", ultimo, nome);
    return 0;
}
```