

Lista 6 (EXERCÍCIOS)

1) Escreva um programa que declare e inicializa, para qualquer valor, um inteiro, um ponto flutuante e um caractere. Imprima o conteúdo, endereço e tamanho das três variáveis.

2) Imprima os nomes e as respectivas idades considerando os dois vetores:

```
int idades[] = {32, 51, 14, 85, 1};
```

```
char *nomes[] = {"Joao", "Jose", "Maria", "Eduardo", "Lisa"};
```

3) Explique a saída do seguinte programa:

```
#include <stdio.h>

int main()
{
    {
        int a[100];
        printf("%X\n", &a);
    }
    {
        char b[100*sizeof(int)];
        printf("%X\n", &b);
    }
    return 0;
}
```

4) Escreva um programa que receba uma palavra do usuário e determine qual o tamanho da palavra (não use strlen).

5) Escreva um programa que receba duas palavras do usuário e determina se elas são iguais (não use strcmp).

- 6) Escreva um programa que receba duas palavras do usuário e crie uma string concatenando as duas (não use strcat).
- 7) Escreva um programa que receba uma frase do usuário e conte a quantidade de palavras.
- 8) Escreva um programa que receba uma palavra e uma frase do usuário, imprima qual a posição da palavra na frase, posição medida em caracteres.
- 9) Modifique o programa anterior para imprimir a posição medida em palavras.
- 10) Crie um programa que receba uma frase do usuário e separe a frase em palavras, utilizando como separador espaços. Múltiplos espaços devem contar como um único separador.
- 11) Faça um programa que receba uma palavra e determine se ela é um palíndromo ou não.
- 12) Faça um programa que encontra o maior e o menor valor em um vetor com 10 inteiros.
- 13) Escreva um programa que ordene um vetor de 10 inteiros por ordem crescente.
- 14) Modifique o anterior para ordem decrescente.
- 15) Faça um programa que calcule a multiplicação de duas matrizes quadradas, o programa deve funcionar para um número arbitrário de elementos.
- 16) Considerando que a função main recebe dois parâmetros “main (int args, char *argv[])”, faça um programa que recebe diversas palavras como entrada pela linha de comando e imprime qual é o maior dos parâmetros. Sugestão de entradas:
exerc.exe
exerc.exe "apenas um"
exerc.exe "maior de todos" "menor" "mais ou menos"

exerc.exe "aaaa" "bbbbbb" "ccccccc" "d" "maior de todos os parametros"

exerc.exe "resposta" "em" "algum lugar" "no" "meio"

17) Descreva o que acontece em todas as linhas do seguinte programa e qual é a saída:

```
#include <stdio.h>

int main()
{
    char a[5][5] = {0};
    printf("%d %d %d\n", sizeof(a), sizeof(a[0]), sizeof(a[0][0]));
    printf("%X %X %d\n", a, a[0], a[0][0]);
    printf("%X %X %d\n", a, a[1], a[1][0]);
    printf("%X %X %X %X %X\n", a[0], a[1], a[2], a[3], a[4]);
    return 0;
}
```

O que mudaria se a matriz “a” fosse alterada de char para int?

18) Qual o resultado do seguinte trecho de código?

```
int valor = 10, *temp, soma = 0;

temp = &valor;
*temp = 20;
temp = &soma;
*temp = valor;
printf("valor = %d, *temp = %d, soma = %d\n", valor, *temp, soma);
```

19) O seguinte código apresenta diversos erros de acesso inválido de memória. Corrija-o.

```

#include <stdio.h>

int main()
{
    char **s;
    char foo[] = "Ola mundo!";
    *s = foo;
    printf("A string: %s\n",s);
    s[0] = foo;
    printf("O caractere s[0]: %s\n",s[0]);
    return 0;
}

```

20) Crie um programa que leia palavras, de até 20 letras, até que o usuário digite um ponto final ou até 20 palavras. Então o programa deverá imprimir as palavras na ordem inversa. Exemplo:

```

um
dois
três
.
três dois um

```

21) Escreva um programa que receba uma frase e troque a primeira letra de cada palavra por uma letra maiúscula, todas as demais letras devem ser minúsculas, mesmo que o usuário entre com maiúsculas.

22) As seguintes linhas de código não deveriam apresentar o mesmo comportamento?

```

int i = 42;

short int *j = (short int *)&i, k1 = *j, k2 = *(j+1);

printf("%d %d\n", *j, *j++);

printf("%d %d\n", k1, k2);

```

Dica: procure por ordem de avaliação na seção 6.5.2.2 do padrão C, procure também pelos termos *type punning* e a regra de *strict aliasing*.