

# Lista de Exercícios – CES-10 – Mês 04

Carlos Henrique Q. Forster

1. (Variáveis locais e globais.) O que é impresso na tela pelo seguinte programa?

```
#include <stdio.h>
int w=1,x=2,y=3,z=4;
void foo(int y)
{
    int z=6;
    extern int w;
    printf("%d %d %d %d ",
           w,x,y,z);
}
```

```
main()
{
    int x=7;
    foo(5);
    foo(x);
    printf("%d %d %d %d.",
           w,x,y,z);
}
```

2. (Variáveis locais e globais.) O que é impresso na tela pelo seguinte programa?

```
#include <stdio.h>
int w=1,x=2,y=3,z=4;
void foo(int y)
{
    static int z=6;
    printf("%d %d %d %d ",
           w,x,y,z);
    z=y;
}
```

```
main()
{
    int x=7, w=8;
    foo(5);
    {
        extern int w;
        foo(w);
    }
    printf("%d %d %d %d.",
           w,x,y,z);
}
```

3. (Chamada de função.) Seja a função **raizes\_reais()** uma função que retorna o número de raízes reais de um polinômio. Os coeficientes do polinômio estariam armazenados num vetor do tipo **float** que é passado como parâmetro **p**, junto com o grau do polinômio (parâmetro **grau**). Deve também ser passado à função um vetor de **float** para que esta retorne o valor das raízes encontradas.

O protótipo da função é o seguinte:

```
int raizes_reais(float p[],
                 int grau, float r[]);
```

Escreva uma função **main** que, chamando a função **raizes\_reais**, imprima na tela todas as raízes reais de um polinômio entrado pelo usuário.

4. (Chamada de função.) Seja a função **linha** a função definida conforme o protótipo abaixo que desenha um segmento de reta na tela gráfica sendo (**x0,y0**) as coordenadas do ponto inicial e (**x1,y1**) as coordenadas do ponto final do segmento.

```
void linha(int x0, int y0,
           int x1, int y1);
```

Defina uma função **retangulo** que, utilizando a função **linha**, desenhe um **retangulo** com lados paralelos à borda da tela com centro em um ponto (**xc,yc**) e **comprimento** e **altura** passados como parâmetro.

5. (Chamada de função.) Seja a função **troca** cujo protótipo está definido abaixo, uma função que permuta o valor de uma variável do tipo **double** por outra, ambas passadas por referência.

Defina uma função **main** que permuta os valores das variáveis **a**, **b** e **c**, declaradas abaixo, de forma que no final  $a \leq b \leq c$ . A função **main** deve chamar a função **troca**.

```
void troca(double*x, double*y);
double a, b, c;
```

6. (Variáveis locais e globais.) O que é impresso pelo seguinte programa?

```
#include <stdio.h>
int x=0;
void main()
{
    int i, x=1;
    printf("%d ",x);
    for(i=0; i<4; i++)
    {
        int x=2;
        printf("%d ",x);
        {
            int x=3;
            printf("%d ",x);
        }
    }
    printf("%d ",x);
}
```

7. (Recursão – simular) Qual o resultado impresso pelo seguinte programa?

```
#include <stdio.h>
void troca(char s[], int p1,
            int p2)
{
    char aux;
    aux=s[p1];
    s[p1]=s[p2];
    s[p2]=aux;
}

void pm(char s[],int k, int n)
{
    int i;
    if (k==n) printf("%s",s);
    else for(i=k; i<n; i++)
    {
        troca(s,k,i);
        pm(s,k+1,n);
        troca(s,k,i);
    }
}
```

```
main()
{
    char t[5]="abcd";
    pm(t,0,4);
}
```

8. (Recursão – simular) Qual o resultado impresso pelo seguinte programa?

```
#include<stdio.h>
int digitsum(int n)
{
    if (n>0)
        return (n%10)
            +digitsum(n/10);
    else
        return 0;
}

int digitalroot(int n)
{
    if(n>9) return
        digitalroot(digitsum(n));
    else
        return n;
}

main()
{
    printf("%d %d %d",
        digitalroot(123),
        digitalroot(1234),
        digitalroot(12345));
}
```

9. (Recursão – simular) Qual o resultado impresso pelo seguinte programa?

```
#include <stdio.h>
int g(int n, int a, int b)
{
    if(n==0) return a;
    else return g(n-1,b,a+b);
}

int f(int n)
{ return g(n,0,1);
}

main()
{ printf("%d",f(8));
}
```

10. (Recursão – construir.) Defina uma função que calcule de forma recursiva o número de combinações de **n** elementos tomados **m** a **m**, utilizando as seguintes regras:

$$\binom{n+1}{m} = \binom{n}{m} \cdot \frac{n+1}{n-m+1} \text{ e}$$
$$\binom{n}{m+1} = \binom{n}{m} \cdot \frac{n-m}{m+1}.$$

11. (Recursão – construir.) Defina uma função para fazer busca binária por recursão em um vetor ordenado. A função é definida pelos seguintes passos: (1) Se o vetor tiver tamanho zero retorne -1 para indicar que o elemento procurado não pertence ao vetor nulo; (2) Teste o elemento central do vetor, se for igual ao elemento procurado retorne seu índice; (3) Se o elemento central for diferente do procurado, procure recursivamente no sub-vetor à esquerda do elemento central e, se ainda não encontrou o elemento procurado, procure-o recursivamente no sub-vetor à direita do elemento central.

12. (Estruturas heterogêneas.) Considere a seguinte estrutura:

```
struct conjunto
{
    int elementos[50];
    int num_elementos;
};
```

Esta estrutura representa um conjunto de elementos. O tamanho do conjunto é dado pelo campo **num\_elementos** e os elementos são representados pelo sub-vetor **elementos[0..num\_elementos-1]**. Admite-se que não existem elementos repetidos. Defina uma função para determinar a intersecção de dois conjuntos representados por esta estrutura.

13. (Recursão – construir.) Defina uma função que efetue potenciação de forma recursiva.

14. (Estruturas heterogêneas.) Defina uma **struct** e outros tipos necessários (vetores, **unions**, **enums** etc) para representar um cadastro de veículos de propaganda (rádio, tv, jornal, outdoor etc). Use sua imaginação para definir os campos.

15. (Estruturas heterogêneas.) Defina uma **struct** e outros tipos necessários (vetores, **unions**, **enums** etc) para representar um catálogo de peças. Use sua imaginação para definir os campos. Defina e inicialize uma variável do tipo estrutura definido.