



Implementar todos os programas em C.

**ESTRUTURAS DE CONTROLE**

- 1) Elabore um programa que leia três valores numéricos quaisquer e os imprima em ordem crescente.
- 2) Crie um programa para ler a nota de um estudante. Se a nota for igual ou acima de 7.0, informar "Estudante aprovado". Se a nota estiver entre 4.0 (inclusive) e 7.0 (exclusive), informar "Estudante em recuperação". E se estiver abaixo de 4.0 informar "Estudante reprovado".
- 3) Tendo como dados de entrada a altura e o sexo de uma pessoa, construa um programa que calcule seu peso ideal utilizando as seguintes fórmulas:
  - para homens:  $(72.7 * h) - 58$ ;
  - para mulheres:  $(62.1 * h) - 44.7$  (sendo  $h$  = altura).
- 4) A Amplitude amostral é uma medida de dispersão, ela é calculada como a diferença entre o valor máximo e o valor mínimo de uma amostra. Elabore um programa que leia um conjunto de 10 valores inteiros e então mostre o valor máximo, o valor mínimo e a amplitude amostral do conjunto fornecido.
- 5) Escreva um programa que leia um conjunto de números inteiros e que somente termine a leitura quando for fornecido um valor 0 (zero) imediatamente após um número ímpar positivo. Informe então qual foi o maior número ímpar fornecido.
- 6) O  $n$ -ésimo número harmônico é dado pelo somatório abaixo.

$$H_n = \sum_{k=1}^n \frac{1}{k}.$$

Escreva um programa para calcular o valor do número harmônico  $H$  dado que o número  $n$  será fornecido pelo usuário. Exemplo, se o usuário digitar 4, então

$$H = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} = 2.08333.$$

- 7) Elabore um programa que calcule o valor da série  $S$  abaixo, sendo que o valor inteiro de  $n$  é fornecido pelo usuário.

$$S(n) = \frac{1}{\sqrt{3}} + \frac{2}{\sqrt{4}} + \frac{3}{\sqrt{5}} + \dots + \frac{n}{\sqrt{n+2}}.$$

- 8) O valor de  $\pi$  pode ser calculado usando como base a seguinte série.

$$S = 1 - \frac{1}{3^3} + \frac{1}{5^3} - \frac{1}{7^3} + \dots + \frac{(-1)^{n+1}}{(2n-1)^3},$$

sendo

$$\pi = \sqrt[3]{32S}$$

Elabore um programa que calcule e mostre o valor de  $\pi$  com base em uma série  $S$  de 50 termos. Compare com o valor exato (3.141592653590).

## FUNÇÕES

- 9) Elaborar uma função que receba 3 números e retorne o menor deles. Testar a função. Exemplo: se os números forem 2, 4 e 9, ela retornará 2.
- 10) Elaborar uma função que receba 3 números e retorne o maior deles. Testar a função. Exemplo: se os números forem 2, 4 e 9, ela retornará 9.
- 11) Elaborar uma função que receba 3 números e retorne o valor intermediário entre eles. Testar a função. Exemplo: se os números forem 2, 4 e 9, ela retornará 4.
- 12) Escreva uma função que calcule o n-ésimo número harmônico (veja exercício 6) . Testar a função.
- 13) Crie uma função que receba por parâmetro um vetor de inteiros e retorne o menor elemento do vetor.
- 14) Escreva uma função que calcule recursivamente o n-ésimo número harmônico (veja exercício 6) . Testar a função.
- 15) Crie uma função recursiva que retorne a soma dos elementos de um vetor de inteiros passado por parâmetro.
- 16) Crie uma função recursiva que receba por parâmetro um vetor de inteiros e retorne o menor elemento do vetor.
- 17) A multiplicação de dois números inteiros pode ser feita através de somas sucessivas. Proponha uma função recursiva `Mult_Rec(n1,n2)` que calcule a multiplicação de dois números inteiros.
- 18) Faça uma função recursiva que receba um número inteiro positivo N e imprima todos os números naturais de 0 até N em ordem decrescente.
- 19) Faça uma função recursiva que receba um número inteiro positivo N e imprima todos os números naturais de 0 até N em ordem crescente.

## ARRANJOS

- 20) Desenvolva um programa que leia 10 números inteiros e armazene-os em um vetor chamado `vLido`. Depois, crie dois outros vetores: `vPares`, contendo somente os números pares de `vLido`, e `vImpares` contendo somente os números ímpares de `vLido`. Os vetores `vPares` e `vLido` não deverão conter zeros. Mostre então os três vetores.
- 21) Desenvolva um programa que leia um vetor de 10 posições reais e o coloque em ordem crescente, utilizando a seguinte estratégia de ordenação:
- selecione o elemento do vetor de 10 posições que apresenta o menor valor;
  - troque este elemento pelo primeiro;
  - repita estas operações, envolvendo agora apenas os 9 elementos restantes (trocando o de menor valor com a segunda posição), depois os 8 elementos (trocando o de menor valor com a terceira posição), depois os 7, 6 e assim por diante, até restar um único elemento, o maior deles.

Observação: este método de ordenação é conhecido como "Seleção Direta".

**22)** Implemente um programa que permita multiplicar uma matriz de ordem (3×3) de números inteiros fornecida pelo usuário por um número também fornecido pelo usuário. A matriz multiplicada deve ser impressa.

Lembrete. Para multiplicar uma matriz  $A_{m \times n}$  por um escalar  $k$ , basta multiplicar cada entrada  $a_{ij}$  de  $A$  por  $k$ . Assim, a matriz resultante  $B$  será também da ordem  $(m \times n)$  e  $b_{ij} = ka_{ij}$ .

**23)** A transposta de uma matriz  $A_{m \times n}$  é a matriz  $A_{n \times m}^T$  em que  $a_{ij}^T = a_{ji}$ , ou seja, os elementos da primeira linha da  $A$  são os elementos da primeira coluna de  $A^T$ , os elementos da segunda linha de  $A$  são os elementos da segunda coluna de  $A^T$ , e assim sucessivamente. Elabore um programa que sorteie as dimensões  $m$  (linha) e  $n$  (coluna) da matriz  $A$  (valores entre 4 a 10, inclusive), preencha então esta matriz com números inteiros (sorteados dentro do intervalo 10 a 99) e calcule  $A^T$ . O programa deve imprimir  $A$  e  $A^T$ .

Como gerar números aleatórios no C. O programa comentado abaixo mostra a geração de números aleatórios dentro de uma faixa de valores (no caso, entre 20 e 49).

```
#include <stdio.h>
#include <stdlib.h> // Necessário para rand(), RAND_MAX e srand().
#include <time.h>   // Necessário para time().
int main(){
    int i,r;
    printf("RAND_MAX=%d.\n",RAND_MAX);
    printf("0<=rand()<=%d.\n",RAND_MAX);
    /* A função srand() lança nova semente para gerar números
    aleatórios distintos a cada execução. */
    srand(time(NULL)); /* Gera semente a partir do relógio. */
    printf("Gerando 10 valores aleatorios entre 20 e 49:\n");
    for (i=0;i<10;i++){
        /* Gera valores aleatórios entre 20 e 49. */
        r=rand()%50; /* r=resto<50 */
        while(r<20) r=rand()%50; /* 20<=r (<50) */
        /* Imprime valores. */
        printf("%d ",r);
    }
}

Resultado do programa
RAND_MAX=32767.
0<=rand()<=32767.
Gerando 10 valores aleatorios entre 20 e 49:
20 23 43 42 41 24 45 25 48 23
```

**24)** Elabore um programa que preencha uma matriz quadrada (4×4) de números inteiros, sorteados dentro do intervalo 100 a 999, garantindo que não haverá nenhuma repetição (os 16 números devem ser únicos). Encontre então o valor do menor elemento da linha em que se encontra o maior elemento da matriz. Mostre a matriz e o valor encontrado.

**25)** Crie uma função que retorne a soma dos elementos de um vetor de inteiros passado por parâmetro.

Exemplo. O programa abaixo mostra como passar um vetor como parâmetro.

```
#include <stdio.h>
// Essa função imprime um vetor até o n-ésimo componente.
void imprime(int vet[],int n){ // n<=tamanho do vetor passado (10).
    int i;
    for(i=0;i<n;i++)
```

```

        printf("%d ",vet[i]);
    }
    // Programa que testa a função imprime().
    int main(){
        // Definição de vetor na declaração.
        int v[]={0,1,2,3,4,5,6,7,8,9}; // Tamanho 10.
        imprime(v,10); // Imprime até o último.
        printf("\n");
        imprime(v,3); // Imprime até o terceiro.
    }

```

**Resultado do programa**

```

0 1 2 3 4 5 6 7 8 9
0 1 2

```

## ESTRUTURAS HETEROGÊNEAS

**26)** Construir um programa que leia o nome, sexo e rendimentos de uma pessoa, guardando os dados em um vetor de struct (exemplo: `rendPessoa RP[10]`, onde `rendPessoa` é um tipo struct). Em seguida, os dados lidos devem ser impressos (ler pelo menos 3 conjuntos de dados). Ao lado, um exemplo de entrada e saída para tal programa. Fazer algo com formatação similar.

```

----jGRASP exec: D:\PI_AtivSoma_02A.exe
>> Digite o nome da pessoa 1.....: Roberto Silva
>> Digite o sexo da pessoa 1.....: M
>> Digite os redimentos da pessoa 1: 3210.39
-----
>> Digite o nome da pessoa 2.....: Mario Guedes
>> Digite o sexo da pessoa 2.....: M
>> Digite os redimentos da pessoa 2: 4322.92
-----
>> Digite o nome da pessoa 3.....: Maria de Fatima
>> Digite o sexo da pessoa 3.....: F
>> Digite os redimentos da pessoa 3: 6007.50
-----

Nome                Sexo    Rend (R$)
-----
Roberto Silva       M       3210.39
Mario Guedes        M       4322.92
Maria de Fatima     F       6007.50
-----
----jGRASP: operation complete.

```

**27)** Uma certa empresa vende tanto imóveis quanto automóveis (bens) em diversos locais. É desejado realizar uma contabilização rápida por período de vendas de cada tipo de bem, sendo que, além do valor de venda, para cada tipo de bem um pequeno conjunto de dados são requisitados para fins de uma breve estatística. Para imóveis é necessário ter sua metragem e o tipo (uma palavra: casa, apto, galpão etc.). Para automóveis, deseja-se ter o ano, número de portas e a potência. Tais informações devem ser guardadas em um único vetor de bens para posterior processamento. Assim, construa um pequeno programa que defina tal vetor, leia os dados para cada

```

Digite o tipo do bem: i
Digite metragem, tipo e valor do imovel:
113 Apto 560000
Digite o tipo do bem: a
Digite o ano, num. de portas, potencia e valor do auto:
2011 4 78 17000

--- BENS FORNECIDOS -----
Imovel:
- Metragem: 113.0 m^2.
- Tipo....: Apto.
- Valor...: R$ 560000.00.
Automovel:
- Ano.....: 2011.
- Portas...: 4.
- Potencia: 78.0 HP.
- Valor...: R$ 17000.00.

```

tipo de bem e, após, imprima o que foi lido. A figura acima mostra um exemplo de execução de tal programa.

Dica. O vetor desejado será um vetor de estruturas semelhante ao apresentado no item "Unões Etiquetadas", mas com a diferença de que a union não conterá tipos básicos, mas duas structs, uma para os dados de automóveis e outra para os de imóveis.