

Funções - Exercícios

Os exercícios a seguir tem como objetivo estimular o uso de funções na linguagem C sob a perspectiva do modelo de entrada-processamento-saída.

PARTE A

Todos os exercícios de **implementação** devem possuir funções e ter as seguintes características:

1. Modelo ENTRADA-PROCESSAMENTO-SAÍDA discutido em sala de aula.
2. As funções devem ser desenvolvidas em duas versões (SAÍDA): estratégia convencional (return) e estratégia do `scanf`. Veja o exemplo a seguir.
3. Representar o estado da memória na chamada das funções. Adote a mesma representação gráfica utilizada nos slides da aula.
4. Fornecer diferentes casos de teste para as funções

Exemplo

Desenvolva um algoritmo para realizar a soma de dois números.

Solução:

```
#include<stdio.h>

int soma1(int a, int b){
    return a + b;
}

void soma2(int a, int b, int *resultado){
    *resultado = a + b;
}

int void main(){
    // Testes de caso 1
    printf("2 + 5 = %d \n", soma1(2,5));
    printf("-5 + 1 = %d \n", soma1(-5,1));

    // Testes de caso 2
    int res;

    soma2(2, 5, &res);
    printf("2 + 5 = %d \n", res);

    soma2(-5, 1, &res);
    printf("-5 + 1 = %d \n", res);
}
```

Exercício 01

Escreva um algoritmo que calcule e mostre a potência de um determinado número somente usando o operador de multiplicação.

Exercício 02

Escreva um algoritmo que dado um número inteiro positivo mostre sua ordem inversa.

Obs.: Não usar string. Utilizar os operadores de divisão e resto.

Exercício 03

O sistema de notas de algumas universidades são calculadas de acordo com a seguinte tabela:

Nota	Conceito
Maior ou igual a 9.0	A
Maior ou igual a 7.0 e menor que 9.0	B
Maior ou igual a 6.0 e menor que 7.0	C
Menor do que 6.0	I (Insuficiente)

Escreva um algoritmo que dado uma nota, converta e mostre o seu conceito equivalente.

Exercício 04

O programa abaixo cria 2 vetores de tamanhos diferentes que são inicializados e modificados durante o código.

Refatore o código utilizando funções e o conceito do modelo entrada-processamento-saída.

```
#include <stdio.h>

int main(){

    int vet1[10];
    int vet2[5];

    int i;
    // Inicializa o vetor com o valor 1
    for (i = 0; i < 10; i++){
        vet1[i] = 1;
    }

    // Imprime o resultado da inicialização
    printf("[");
    for (i = 0; i < 10; i++){
        printf("%d", vet1[i]);
        if(i < 9) printf(",");
    }
    printf("]\n");

    // Multiplica todos os valores por -1
    for (i = 0; i < 10; i++){
        vet1[i] *= -1;
    }

    // Imprime o resultado da multiplicação
    printf("[");
    for (i = 0; i < 10; i++){
        printf("%d", vet1[i]);
        if(i < 9) printf(",");
    }
    printf("]\n");

    // Inicializa o vetor VET2 com o valor 20
    for (i = 0; i < 5; i++){
        vet2[i] = 20;
    }

    // Imprime o resultado da inicialização
    printf("[");
    for (i = 0; i < 5; i++){
        printf("%d", vet2[i]);
        if(i < 4) printf(",");
    }
    printf("]\n");

    // Multiplica todos os valores por 2
    for (i = 0; i < 5; i++){
        vet2[i] *= 2;
    }

    // Imprime o resultado da inicialização
```

```
    printf("[");
    for (i = 0; i < 5; i++){
        printf("%d", vet2[i]);
        if(i < 4) printf(",");
    }
    printf("]\n");
}
```

Exercício 05

O programa abaixo realiza a leitura de uma matriz de 3 x 3 elementos, multiplica cada elemento por 5 e imprime o resultado. Faça uma refatoração do código utilizando funções e o conceito do modelo de entrada-processamento-saída. Identifique oportunidades de generalizar e reaproveitar o código. Mantenha uma responsabilidade por função.

Sugestão:

- Função para leitura
- Função para impressão
- Função para multiplicação da matriz

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

int main(){
    int i, j, m[3][3];
    //captura os elementos
    for(i=0;i<3;i++){
        for(j=0;j<3;j++){
            printf("Elemento[%d][%d]= ",i,j);
            scanf("%d",&m[i][j]);
        }
    }
    //EXIBIR VALORES ORIGINAIS
    printf("\n::: Valores Originais :::\n");
    for(i=0;i<3;i++){
        for(j=0;j<3;j++)
            printf("%d ",m[i][j]);
        printf("\n");
    }

    //multiplica por 5
    for(i=0;i<3;i++)
        for(j=0;j<3;j++)
            m[i][j]=m[i][j]*5;

    //EXIBIR VALORES MULTIPLICADOS
    printf("\n::: Valores Multiplicados por 5:::\n");
```

```
    for(i=0;i<3;i++){
        for(j=0;j<3;j++){
            printf("%d ",m[i][j]);
            printf("\n");
        }
    }
    return 0;
}
```

Exercício 06

Escreva um algoritmo para somar todos os valores de um vetor de inteiros.

Exercício 07

Crie uma função para encontrar o maior elemento de um vetor.

Exercício 08

Considerando que temos dois vetores ORDENADOS de números inteiros, o vetor v1 e o vetor v2, desenvolva um algoritmo para intercalar esses dois vetores de modo que o vetor resultante também fique ordenado.

Não usar nenhum algoritmo de ordenação.

PARTE B

Nos exercícios da parte B não há necessidade de desenvolver 2 versões das funções.

Exercício 09

Escreva uma função que devolva a quantidade de caracteres de uma string. **Não é permitido utilizar funções da biblioteca string.h.**

Exercício 10

Escreva uma função que inverta o conteúdo de uma string. Por exemplo: UTFPR => RPFTU **Não é permitido utilizar funções da biblioteca string.h.**

Exercício 11

Escreva uma função que devolva a quantidade de vogais em um string. **Não é permitido utilizar funções da biblioteca string.h.**

Exercício 12

Escreva uma função que copie uma string para outra, tal como o strcpy. **Não é permitido utilizar funções da biblioteca string.h.**
