

## Laboratório 3

### Constantes;

**Desvio condicional (if-else, switch-case, ternário).**

### **Introdução à Programação C (CMT012)**

**Prof. Ronald Souza**

IC/UFRJ — 30/08/2023

Turma: 12422

Google Classroom: 4v5wlvf

### **Objetivo**

Praticar os conceitos básicos de programação vistos na Aula 3.

**Todos os seus programas devem preencher a estrutura abaixo.**

```
/*
Autor: <nome do aluno>
Data: <data de hoje>
Descrição: <o que o programa faz>
Entrada: <o que o programa espera receber como entrada>
Saída: <o que o programa retorna para o usuário>
Defesa: <restrições sobre os dados de entrada, se
existirem>
*/

#include <stdio.h>

int main() {
    //dicionário de dados

    //corpo do programa

    return 0;
}
```

## RELEMBRANDO

### Para iniciar o computador no LEP-1 - Linux:

1. Ligue o computador e escolha o sistema operacional Linux.
2. Use para **login** e **senha** a palavra “convidado”.
3. Abra um **Terminal** para digitar comandos.
4. Ao abrir a janela do terminal, você estará dentro do diretório (ou pasta) raiz do seu usuário (ex., */home/convidado*). Digite **pwd** para checar.
5. Crie um diretório para guardar seus programas: **mkdir prog1**
6. Vá para o diretório criado digitando **cd prog1**
7. Verifique quais arquivos o diretório contém, digitando **ls** (nesse primeiro momento estará vazio).

### Para escrever um programa em C:

1. Abra um editor de texto simples qualquer (ex.: **gedit** ou **kate**);
2. Digite o seu programa e salve-o no diretório */home/convidado/prog1/*
3. Dê um nome ao seu arquivo, **lembrando que a extensão deve ser “.c”** Por exemplo, nosso primeiro programa recebeu o nome **aloMundo.c**

### Para compilar e executar um programa C:

1. **Compile** na janela do Terminal, no diretório */home/convidado/prog1/* digitando o seguinte comando:  
`gcc <nome do prog>.c -o <nome do prog>.out -Wall`  
Por ex., se o seu programa se chama **aloMundo.c**, o comando será:  
`gcc aloMundo.c -o aloMundo.out -Wall`
2. **Execute** digitando: `./<nome do prog>.out`  
Por exemplo:  
`./aloMundo.out`

### Atividade 1 - Leia com calma esta página e execute a tarefa da página seguinte.

Em aula vimos a estrutura básica de um comando switch-case, ilustrada no exemplo abaixo:

```
int main() {
    int voto;
    printf("Digite o seu candidato.\n");
    scanf ("%d", &voto);
    switch(voto) {
    case 15:
        printf("Candidato(a) A.\n");
        break;
    case 29:
        printf("Candidato(a) B.\n");
        break;
    case 40:
        printf("Candidato(a) C.\n");
        break;
    default:
        printf("Voto anulado.\n");
    }
    return 0;
}
```

Poderíamos ter utilizado “constantes”\* para melhor legibilidade:

```
#define PARTIDO_X 15
#define PARTIDO_Y 29
#define PARTIDO_Z 40
int main() {
    int voto;
    printf("Digite o seu candidato.\n");
    scanf ("%d", &voto);
    switch (voto) {
    case PARTIDO_X:
        printf("Candidato(a) A.\n");
        break;
    case PARTIDO_Y:
        printf("Candidato(a) B.\n");
        break;
    case PARTIDO_Z:
        printf("Candidato(a) C.\n");
        break;
    default:
        printf("Voto anulado.\n");
    }
    return 0;
}
```

\*Na cláusula *case* **não** se admite o uso de constantes *verdadeiras* (isto é, variáveis com o modificador de acesso **const**), pois em C **toda cláusula case requer um literal**.

**Tarefa:** Vimos que é possível avaliar qualquer expressão logicamente interpretável (ou seja, expressões que resultam em um número inteiro). Por exemplo, um tipo **char** como argumento do `switch()` gera uma expressão válida, pois será interpretado seu código ASCII (um número inteiro):

```
int main() {
    char simbolo;
    printf("Digite um caractere.\n");
    scanf ("%c", &simbolo);
    switch(simbolo) {
    case '!':
        printf("Exclamacao.\n");
        break;
    case '?':
        printf("Interrogacao.\n");
        break;
    case '@':
        printf("O proximo simbolo eh %c.\n", simbolo + 1);
    default:
        printf("Simbolo qualquer.\n");
    }
    return 0;
}
```

→ **Rode o código acima** e digite o símbolo '@'. Por que foi exibida a letra A e a mensagem padrão?

→ **Corrija** o exemplo para que a mensagem padrão **não** seja exibida nesse caso.

## Atividade 2

Vimos em aula o uso básico do operador ternário

`(<expr avaliada>) ? <expr se V> : <expr se F>`

ilustrado no exemplo abaixo:

```
int main(void) {
    int x, y;
    int maior;

    printf("Digite dois inteiros:\n");
    scanf("%d %d", &x, &y);

    maior = (x > y) ? x : y;

    printf("O maior valor eh: %d", maior);
    return 0;
}
```

Operadores ternários podem ser *aninhados*. Por exemplo, na estrutura mais acima, podemos substituir `<expr se V>` por:

`(<expr avaliada>) ? (<expr 2>) ? <expr se V> : <expr se F> : <expr se F>`

O mesmo poderia ser feito com a `<expr se F>`.

→ **Use o comando ternário para implementar as seguintes operações em C:**

- (a) receber um inteiro e retornar -1 se ele for par e 1 se ele for ímpar;
- (b) receber um racional e retornar 1 se ele for positivo e 0 caso contrário;
- (c) receber três racionais e retornar o maior deles;
- (d) receber três racionais e retornar a média aritmética dos dois maiores;
- (e) receber um caractere e retornar:
  - 'm' se ele for uma letra **minúscula** do alfabeto;
  - 'M' se ele for uma letra **maiúscula** do alfabeto;
  - '?' se for um outro caractere qualquer.

### Atividade 3

Escreva um programa em C que oferece um menu de opções **via switch-case** para executar as operações definidas na atividade anterior (na mesma ordem que elas foram passadas). Declare uma pseudo-constante para cada opção (via `#define`, como no exemplo da pág. 3). Acrescente uma opção para sair do programa. Caso o usuário entre com uma opção inválida, o programa deverá informar isso e terminar.

**Atividade 4:** Escreva um programa que receba duas notas e compare a média aritmética destas com um **valor constante igual a 5**, que representa o limiar de aprovação. Caso a média seja maior ou igual ao limiar, informar “Aprovado”. Imprimir “Reprovado” caso contrário. *Para a declaração da constante, utilize o modificador **const** em uma variável.*

**Atividade 5:** Substituir o comando de seleção **if-else** pelo **switch-case** no trecho de código abaixo. **Teste o seu programa e verifique a validade das operações.**

```
#include <stdio.h>
int main() {
    float a, b;
    char operador;

    printf("Digite uma operacao aritmetica entre dois numeros a e b, no formato \"a
<operacao> b\":\n");

    scanf("%f %c %f", &a, &operador, &b);

    if (operador == '+') {
        printf("%.2f\n", a + b);
    }
    else if (operador == '-') {
        printf("%.2f\n", a - b);
    }
    else if (operador == '/') {
        printf("%.2f\n", a / b);
    }
    else if (operador == '*') {
        printf("%.2f\n", a * b);
    }
    else {
        puts("Operador nao definido.");
    }
    return 0;
}
```