

# PARTE III \_ SELETORES

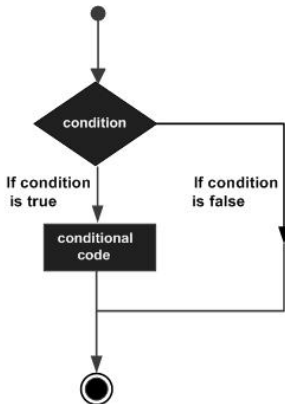
Marcos Silvano / DACOM

BCC32A-Algoritmos 1

- Comandos de seleção e exemplos
  - ▶ IF
  - ▶ IF ELSE
  - ▶ IF ELSE aninhados
  - ▶ IF ELSE encadeados
  - ▶ SWITCH
  - ▶ SWITCH aninhados
- Baseado no curso sobre linguagem C do site Tutorials Point
  - ▶ <https://www.tutorialspoint.com/cprogramming/>

# Seletores/Seleção

- A estrutura de seleção ou decisão é utilizada para decidir se um bloco de comandos será executado (ou não).
  - ▶ O bloco interno é executado somente se a **condição for verdadeira**
  - ▶ O código que encontra-se após o condicional, **continuará a ser executado**



# Seleção IF

```
if (condition) {  
    statement1;  
    statement2;  
    statement3;  
    ...  
}
```

- Os comandos (statements) são executados se a condição (condition) for verdadeira. Uma condição é uma expressão booleana que resulta em dois valores possíveis: verdadeiro (true ou 1) ou falso (false ou 0).
  - ▶ Em C/C++ o valor 0 é **false** e qualquer outro, é considerado **true**
- Em C/C++, torna-se obrigatória a utilização de bloco (chaves) quando existe mais de um comando a executar.
  - ▶ No esquema acima, os comandos entre chaves { } só serão executados se a condição for verdadeira.

# Seleção IF: Exemplo

```
#include <stdio.h>

int main () {
    int a;

    print("Informe o valor: ");
    scanf("%d", &a);

    /* verifica condição */
    if( a > 50 ) {
        /* se condição for verdadeira, executa este bloco de código */
        printf("Maior que 50\n" );
    }

    printf("Valor: %d\n", a);

    return 0;
}
```

# Seleção IF ELSE

```
if (condition) {  
    /* block 1 */  
    statement1;  
    statement2;  
} else {  
    /* block 2 */  
    statement3;  
    statement4;  
}
```

- Se a condição for verdadeira, **block 1** será executado.
- Caso contrário (condição é falsa), **block 2** será executado.

# Seleção IF ELSE: Exemplo

```
#include <stdio.h>

int main () {
    int a;

    print("Informe o valor: ");
    scanf("%d", &a);

    /* verifica condição */
    if( a > 50 ) {
        /* se condição for verdadeira, executa este bloco de código */
        printf("Maior que 50\n" );
    } else {
        /* se condição for verdadeira, executa este bloco de código */
        printf("Menor que 50\n" );
    }

    printf("Valor: %d\n", a);

    return 0;
}
```

# Seleção IF ELSE aninhados

- Podemos “aninhar” código dentro de um bloco, incluindo outro comando de seleção.

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

```
int main () {
```

```
    int a;
```

```
    int b;
```

```
    print("Informe os valores: ");
```

```
    scanf("%d %d", &a, &b);
```

```
    /* verifica condicao booleana */
```

```
    if( a == 100 ) {
```

```
        /* se a primeira condicao é verdadeira, verifica esta */
```

```
        if( b == 200 ) {
```

```
            /* se ambas condições são verdadeiras, executa este bloco */
```

```
            printf("A == 100 e B == 200\n" );
```

```
        }
```

```
    }
```

```
    printf("A: %d e B: %d\n", a );
```

```
    return 0;
```

```
}
```



# Seleção IF ELSE aninhados

- Em alguns casos, podemos substituir seletores aninhados por expressões booleanas compostas.

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

```
int main () {  
    int a;  
    int b;
```

```
    print("Informe os valores: ");  
    scanf("%d %d", &a, &b);
```

```
    /* operador lógico AND (&&) resultad verdadeiro se ambos operandos forem verdadei  
    if( a == 100 && b == 200 ) {  
        /* se ambas condições são verdadeiras, executa exste bloco */  
        printf("A == 100 e B == 200\n" );  
    }
```

```
    printf("A: %d e B: %d\n", a );
```

```
    return 0;
```

```
}
```

# Seleção IF ELSE aninhados

- Outro exemplo: encontrar o maior entre 3 valores.
- Como ficaria este exemplo usando **expressões booleanas compostas**?

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

    printf("Informe os valores: ");
    scanf("%d %d %d", &a, &b, &c);

    if (a > b) {
        if (a > c) { // a > b > c
            printf("A é MAIOR\n");
        } else { // c > a > b
            printf("C é MAIOR\n");
        }
    } else {
        if (b > c) { // b > a > c
            printf("B é MAIOR\n");
        } else { // c > b > a
            printf("C é MAIOR\n");
        }
    }
    return 0;
}
```

# Seleção IF ELSE-IF: Encadeamentos

- Em alguns casos queremos selecionar **APENAS UMA** das opções.
- O que acontece neste código?

```
#include <stdio.h>
int main() {
    int score;
    printf("Informe a nota (0-10): ");
    scanf("%d", &score);

    if (score >= 9) {
        printf("Conceito: A\n");
    }
    if (score >= 8) {
        printf("Conceito: B\n");
    }
    if (score >= 7) {
        printf("Conceito: C\n");
    }
    if (score >= 6) {
        printf("Conceito: D\n");
    }
    else {
        printf("Conceito: F\n");
    }
    return 0;
}
```

# Seleção IF ELSE-IF: Encadeamentos

- Seleccionar uma opção => **encadeamentos de IF ELSE-IF**

```
#include <stdio.h>
int main() {
    int score;
    printf("Informe a nota (0-10): ");
    scanf("%d", &score);

    if (score >= 9) {
        printf("Conceito: A\n");
    }
    else if (score >= 8) {
        printf("Conceito: B\n");
    }
    else if (score >= 7) {
        printf("Conceito: C\n");
    }
    else if (score >= 6) {
        printf("Conceito: D\n");
    }
    else {
        printf("Conceito: F\n");
    }
    return 0;
}
```

# Seleção SWITCH

- Caso queiramos um encadeamentos de seletores para **comparar a igualdade um valor**, podemos usar o comando SWITCH
- Sempre podemos usar IF ELSE-IF no lugar do SWITCH
- Para comparar intervalos de valores, somente IF ELSE-IF
- Qual o motivo do **break** ao final de cada **case**?

```
#include <stdio.h>
int main() {
    int score;
    printf("Informe um número (0-5): ");
    scanf("%d", &score);

    /* select only one option */
    switch (score) {
        case 0: printf("Number: Zero\n"); break;
        case 1: printf("Number: One\n"); break;
        case 2: printf("Number: Two\n"); break;
        case 3: printf("Number: Three\n"); break;
        case 4: printf("Number: Four\n"); break;
        case 5: printf("Number: Five\n"); break;
        default: printf("Error: not in range (0-5)\n");
    }
    return 0;
}
```

# Seleção SWITCH

- Em alguns momentos, desejamos que mais de uma condição seja selecionada para um valor

```
int main () {
    char grade;
    printf("Informe o conceito (A-F): ");
    scanf("%c", &grade);
    switch(grade) {
        case 'A' :
            printf("Excelente!\n" );
            break;
        case 'B' :
        case 'C' :
            printf("Muito bem!\n" );
            break;
        case 'D' :
            printf("Você passou\n" );
            break;
        case 'F' :
            printf("Melhor tentar novamente\n" );
            break;
        default :
            printf("Conceito inválido\n" );
    }
    printf("Conceito  %c\n", Conceito );
}
```

# Seleção SWITCH: Aninhamentos

- Conforme mencionado anteriormente, “aninhar” qualquer código dentro de um bloco.

```
#include <stdio.h>

int main () {
    int a;
    int b;

    printf("Informe os valores: ");
    scanf("%d %d", &a, &b);

    switch(a) {

        case 100:
            printf("Switch externo\n", a );

            switch(b) {
                case 200:
                    printf("Switch interno\n", a );
            }
    }
    printf("A: %d\n", a );
    printf("B: %d\n", b );
    return 0;
}
```

# Alguns programas para implementar

- Programa [1]

- ▶ Lê caractere da entrada. Se não for letra, informa. Se for letra, verifica se é maiúscula (e passa para maiúscula se necessário).

- Programa [2]

- ▶ Calculadora simples: recebe valores A e B, bem como operador. Mostra expressão e resultado.

- Programa [3]

- ▶ Lê inteiro representando ano e verifica se é ano bissexto. Para um ano ser bissexto (leap year), ele precisa:
  - ★ Ser divisível por 4 e não divisível por 100, ou;
  - ★ Ser divisível por 400.

- Programa [4]

- ▶ Lê caractere e informa:
  - ★ Se é letra e, neste caso, também informa se é vogal ou consoante
  - ★ Se é número
  - ★ Se é símbolo