

Revisão I – Comandos de Decisão e de Repetição

Parte 1. Comandos de Decisão

1. Comandos if/else

Usado para tomar decisões com base em condições:

1. Executa um bloco de comandos quando a expressão for VERDADEIRA (1);
2. ou outro bloco de comandos quando uma expressão for FALSA (0).

Sintaxe/Declaração geral:

```
if (condição) {  
    // Executa esse comando/blocos de comandos se for verdadeira  
  
} else { // Se for FALSA,  
    // se a anterior falhar, executa esse código/blocos de comandos  
}
```

Exemplo 01: No Brasil, quando uma pessoa é classificada como idosa (60 anos ou mais), tem direito à gratuidade no uso do transporte público. Escreva um código que, dada a idade de uma pessoa, informe se ele tem direito à gratuidade ou não.

```
#include <estádio>  
int main() {  
    int idade;  
  
    printf("Entre com a idade da pessoa: \n");  
    scanf("%d", &idade);  
  
    if (idade >= 60) {  
        printf("Tem direito à gratuidade!\n");  
    }  
    else {  
        printf("Não tem direito à gratuidade!\n");  
    }  
    return 0;  
}
```

2 Comandos if/else if

Permite selecionar **apenas uma opção** (um comando/bloco de comandos) dentre **várias opções disponíveis** dentro do programa.

Sintaxe/Declaração geral:

```
if (condição) {  
    // Executa esse código/blocos de comandos se verdadeira  
  
} else if (outra_condição) {  
    // Executa esse código/blocos de comandos se anterior for falsa e essa verdadeira  
  
} else {  
    // Se todas as anteriores falharem, executa esse código/blocos de comandos  
  
}
```

Exemplo 02: Programa que lê altura e o peso de uma pessoa e calcula seu IMC. Em seguida o programa classifica a pessoa de acordo com a Tabela 1.

Tabela 1: Classificação IMC

Condição	IMC em Adultos
Abaixo do peso	$< 18,5$
Peso normal	$18,5 \leq \text{IMC} \leq 24,9$
Sobrepeso	$25 \leq \text{IMC} \leq 29,9$
Obeso	≥ 30

```
#include <stdio.h>
int main() {
    float peso, altura, imc;

    printf("Digite o peso (kg): ");
    scanf("%f", &peso);
    printf("Digite a altura (m): ");
    scanf("%f", &altura);

    imc = peso / (altura * altura);

    if (imc < 18.5) {
        printf("Abaixo do peso\n");
    } else if (imc < 25) {
        printf("Peso normal\n");
    } else if (imc < 30) {
        printf("Sobrepeso\n");
    } else {
        printf("Obeso\n");
    }
    return 0;
}
```

3. Switch Case

Mais organizado que muitos if else/if, quando lidamos com valores fixos (como números ou caracteres).

Sintaxe/Declaração geral:

```
switch (variavel) {
    case opção1:
        // executa comando/blocos de comando 1
        break;
    case opção2:
        // executa comando/blocos de comando código 1
        break;
    default: // se todas as opções anteriores falharem
        // executa comando/blocos de comando 3
}
```

4 Operador Ternário

Opção **compacta** de if else. Ideal para expressões simples.

Sintaxe/declaração geral:

condição ? expressão1_se_verdadeiro : expressão2_se_falso;

Exemplo 03: Dado um número inteiro, informa se ele é par ou ímpar.

```
#include <stdio.h>
int main() {
    int numero;

    printf("Entre com um número inteiro: ");
    scanf("%d", &numero);

    (numero % 2 == 0) ? printf("O número %d é par\n", numero):
                      printf("O número %d é ímpar \n", numero);

    return 0;
}
```

Para Praticar

1. Nos parques de diversão, alguns brinquedos tem idade e altura mínimas para poder andar neles. Suponha que um determinado parque possui 3 brinquedos com as seguintes limitações (Paes, 2016):

- Barca Viking: 1,5m de altura e 12 anos
- Elevator of Death: 1,4m de altura e 14 anos
- Final Killer: 1,7m de altura ou 16 anos

Dada a altura (em cm) e a idade de uma pessoa, faça um programa que identifique quantos brinquedos ela pode andar.

2. Em uma corrida os participantes partem de um mesmo local e tentam chegar ao mesmo destino. Vence quem chegar em menos tempo. Cada opção de caminho é dada por um número, conforme mostra a Figura 1. Todos os participantes foram até o final da trilha, mas alguns fizeram um caminho inválido (ex., 3, 4 e 6). Faça um programa que receba um determinado caminho e indique se este é válido e se a equipe chegou ao destino. Cada caminho é representado por uma sequência de números (Paes, 2016).

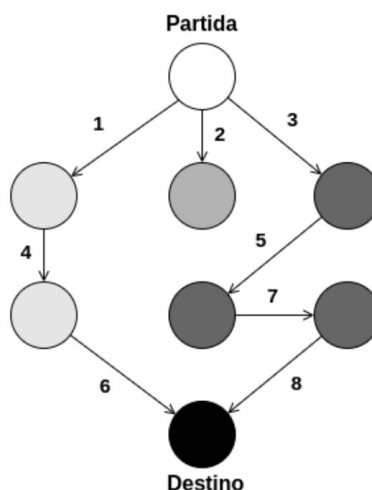


Figura 1: Mapa da corrida

Parte 2: Comandos de Repetição

1. Comando `while`

Repete enquanto a condição for verdadeira. **Checa antes** de executar.

Sintaxe/declaração geral:

```
while (condição) {  
    // repete comando/blocos de comando enquanto condição for verdadeira  
}
```

Exemplo 04: Crie um código usando for para imprimir somente os números ímpares de 1 até 20.

```
#include <stdio.h>  
int main() {  
    int impar;  
  
    impar = 1;  
    while (impar <= 20) {  
        printf ("%d\n", impar);  
        impar += 2;  
    }  
    return 0;  
}
```

Para Praticar

3. Faça um programa que receba vários números inteiros positivos digitados pelo usuário e ao final exibe a quantidade de números digitados e a soma total desses números. O programa deve ser encerrado quando o usuário digitar 0.

2, Comando do while

Garante uma execução, depois verifica a condição.

Sintaxe/declaração geral:

```
do {  
    // executa comando/blocos de comando pelo menos uma vez  
} while (condição);
```

Exemplo 05: Crie um código usando for para imprimir somente os números ímpares de 1 até 20.

```
#include <stdio.h>  
int main() {  
    int impar;  
  
    impar = 1;  
    do {  
        printf ("%d\n", impar);  
        impar += 2;  
    } while (impar <= 20);  
    return 0;  
}
```

Para Praticar

4. Faça um programa que receba um número secreto (fixo ou aleatório) e o usuário tenta adivinhar que número é esse. O programa termina quando o usuário acerta, mostrando quantas tentativas ele precisou. A cada tentativa o programa deve informar a seguinte mensagem:

- "Errou! Tente um número maior": se o chute for menor.

- "Errou!" Tente um número menor": se o chute for maior.
- "Acertou!" Você precisou de **n** tentativas".: se o usuário acertou.

3. Comando for

Ideal quando sabemos **quantas vezes** queremos repetir.

Sintaxe/declaração geral:

for (inicialização; condição; incremento) {

```
// executa comando/blocos de comando de comandos
}
```

Exemplo 06: Crie um código usando for para imprimir somente os números ímpares de 1 até 20.

```
#include <stdio.h>
int main() {
    int impar;

    for(impar = 1; impar <= 20; impar +=2) {
        printf ("%d\n " , impar);
    }
    return 0;
}
```

Para Praticar

5 Faça um programa que receba vários números inteiros positivos do usuário, em seguida, o programa deve somar o primeiro, subtrair o segundo, somar o terceiro, subtrair o quarto, e assim por diante. A quantidade de números deve ser fornecida pelo usuário.

Exemplo:

10 - 4 + 3 - 4 + 2 = 7

6. Faça o Teste de Mesa do programa a seguir.

```
#include <stdio.h>
int main() {
    int i, j, count = 0;

    for (i = 1; i <= 4; i++) {
        for (j = 1; j <= 4; j++) {
            if ((i + j) % 3 == 0) {
                count++;
            }
        }
    }
    printf("%d \n", count);
    return 0;
}
```

i	j	i + j	(i + j) % 3 == 0)	count

[illegible]

Referências

PAES, R. B. Introdução à programação com a linguagem C. São Paulo: Novatec, 2016.
PINHEIRO, . A. C. Elementos de Programação em C. Bookman, 2012.