

Programação de Computadores I

Programa Principal e Estruturas Condicionais.

Programa Principal

O ponto de início de um programa C é a função *main()*, ela é obrigatória, ou seja, se ela não existir um erro será retornado porque sistema operacional não saberá por onde começar a execução.

Estrutura básica:

```
int main ( ) {  
    // seu código  
    return 0;  
}
```

Retorna um inteiro (int) para o sistema operacional para informar se a execução foi correta, quando retorna 0, ou se algum erro ocorreu (1).

Estruturas Condicionais

- As estruturas condicionais em C permitem que o programa tome decisões e execute diferentes blocos de código com base em condições lógicas (verdadeiro ou falso). As principais estruturas condicionais são:
 - If – else;
 - Operador ternário ?; e
 - switch.

if

Formato:

```
if (condição) {  
    bloco de código  
}
```

A instrução if permitirá a execução do bloco de código associado a ela se a condição for verdadeira; caso a condição seja falsa então o bloco de código não será executado, e a instrução após esse bloco que será executado.

Exemplo:

```
if (idade < 12) {  
    printf("Criança \n");  
}
```

if-else

Formato:

```
if (condição) {  
    bloco de código 1  
}  
else {  
    bloco de código 2  
}
```

A instrução if permitirá a execução do bloco de código 1, associado a ela, se a condição for verdadeira; caso a condição seja falsa então o bloco de código 2 será executado.

Operador ternário ?

Formato:

(condição) ? valor_se_verdadeiro : valor_se_falso

Uma forma compacta de escrever um if-else simples em uma única linha. Não executa um bloco de código apenas retorna um valor.

Exemplo:

```
int nota=7;  
printf( " %s \n", (nota >6)? "Aprovado" : "Reprovado");
```

switch

Ideal para testar uma única variável contra vários valores constantes específicos (números ou caracteres). É mais eficiente e organizado que múltiplos if encadeados.

Formato:

```
switch( variável) {  
    case valor1: bloco de código 1  
        break;  
    case valor2: bloco de código 2  
        break;  
    ...  
    default : último bloco de código  
}
```