



1. Controle de Fluxo

Os comandos de controle de fluxo são divididos em: estruturas condicionais, estruturas de repetição e os desvios incondicionais.

1.1) Instruções Condicionais

As instruções condicionais existentes na linguagem C (padrão ANSI) são: **if**, **switch** e o operador ternário (?).

Instrução if

Esta instrução obedece à sintaxe:

```
if (<condição>) {  
    bloco de instruções para condição verdadeira ;  
}  
else {  
    bloco de instruções para condição falsa;  
}
```

A presença do caminho **else** é opcional conforme mostra o exemplo abaixo para verificar se um número é par:

```
#include <stdio.h>  
  
int main(){  
    int num;  
  
    printf("Digite um valor inteiro:");  
    scanf("%d", &num);  
  
    if ((num%2)==0) {  
        printf("O numero eh par\n");  
    }  
}
```

No exemplo acima temos a expressão **((num%2)==0)** que verifica se um dado número é par. Essa verificação se dá utilizando-se o operador aritmético **%** que calcula o resto da divisão entre dois números, e o operador relacional **==** que compara duas expressões ou valores e retorna 0 (falso) ou 1 (verdadeiro).

A instrução **if** pode conter novas instruções, que podem ser **if** aninhados, como mostra o trecho no exemplo abaixo:

```
int main(){  
    int num;  
  
    printf("\n Entre com um numero inteiro: ");  
    scanf("%d", &num);  
    if(num >= 0) {  
        if(num%2==0) {  
            print("O numero eh par e positivo.\n"); }  
        else {  
            print("\O numero eh impar e positivo.\n"); }  
    }  
    else {
```

```

if(num%2==0) {
    print("O numero eh par e negativo.\n"); }
else {
    print("O numero eh impar e negativo.\n"); }
}

```

Obs: para que os acentos sejam impressos pela função *printf*, podem-se usar os valores correspondentes às letras acentuadas obtidas na tabela ASCII. Por exemplo, o valor correspondente à letra “e” com acento agudo é 130:

```
printf("O numero %c impar e negativo.\n",130);
```

O operador ternário (?)

O operador ? pode ser utilizado no lugar da instrução if. Este operador requer três operandos e pode ser escrito da seguinte forma:

Exp1? Exp2: Exp3

O exemplo a seguir mostra um teste baseado no valor de x: se o resto da divisão de x por 2 for 0, então o valor 1 deve ser escolhido, senão escolhe-se o valor 0. O valor escolhido (ou calculado) é, ao final, armazenado na variável y.

```

#include <stdio.h>

int main(){
    int x,y;

    printf("Informe um numero inteiro: ");
    scanf("%d",&x);

    y= x%2==0 ? 1 : 0;
    printf("Resultado em Y (1-Par, 0-Impar) : %d\n",y);
}

```

Instrução SWITCH

A instrução **switch** é uma instrução condicional de múltiplos desvios. A instrução analisa o valor resultante de uma expressão ou variável, e de acordo com o valor obtido desvia a execução dos programas para um dos casos possíveis.

Sintaxe:

```

switch ( <expressao> )
{
    case <valor1>: <sequencia 1 de instruções a serem executadas>
                break;
    case <valor2>: <sequencia 2 de instruções a serem executadas>
                break;
    ...
    case <valorn>: <sequencia n de instruções a serem executadas>
                break;
    default: <sequencia a ser executada caso nenhuma anterior seja adequada>
            break;
}

```

Exemplo de uma pequena calculadora em C, usando o comando switch:

```
#include<stdio.h>

int main(){
    float v1, v2;
    char op;

    printf("Entre com valor 1: ");
    scanf("%f", &v1);
    printf("Entre com valor 2: ");
    scanf("%f", &v2);
    printf("Entre com o operador: ");
    scanf("%c", &op);

    switch(op) {
        case '+':
            printf("%.2f + %.2f = %.2f\n", v1, v2, v1 + v2 );
            break;
        case '-':
            printf("%.2f - %.2f = %.2f\n", v1, v2, v1 - v2 );
            break;
        default: printf("Operador invalido: \n");
    }
}
```

Exercícios em linguagem C:

- 1) Faça um programa que receba dois números e mostre qual deles é o maior.
- 2) Analise e teste o código abaixo para validar uma data. Corrija o programa incluindo o teste que verifica se o ano é bissexto e atualiza o valor do dia correspondente ao mês de fevereiro.

```
#include <stdio.h>
#include <stdlib.h>
#define MESES 12

int main(){
    int ndias[MESES] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
    int i, dia, mes, ano;

    printf("Informe o dia, mes e ano: ");
    scanf("%d %d %d",&dia,&mes,&ano);

    mes=mes-1;

    if ((mes>=0) && (mes < MESES) && (dia<=ndias[mes]) && (dia>0)) {
        printf("\nData Valida\n");
    }
    else {
        printf("\nData Invalida\n");
    }
    printf("\n\nMeses/Dias do Ano");
    for(i=0;i<MESES;i++){
        printf("\nMes %d -> %d",i+1,ndias[i]);
    }
}
```

3) Re-escreva a instrução `a = (a > 5) ? 10 : 50;` usando `if`:

4) Usando o comando `switch`, construir um programa para obter aleatoriamente um número de 1 a 7. Em seguida o programa deve informar o dia da semana correspondente, sendo domingo o dia de número 1.

5) Usando o comando `switch`, elaborar um programa para ler um número de 1 a 10, informado pelo usuário, e exibir o número por extenso. Para números fora do intervalo, mostrar uma mensagem de erro.

6) Completar o exemplo da calculadora em linguagem C incluindo os operadores de multiplicação, divisão e exponenciação.

7) Seja o exemplo abaixo para verificar se um número é primo. O cálculo do número primo utilizou uma instrução de repetição (`for`). Mantenha a instrução `for`, modificando apenas a quantidade verificações para `numero/2`. Teste a função antes e depois e verifique se houve alteração significativa no tempo de processamento. Utilize a instrução `switch` (multidirecional) para substituir o comando de seleção encontrado.

```
int main(){
    int ehPrimo, ind, numero;

    printf("Informe o numero a ser verificado: ");
    scanf("%d",&numero);

    ehPrimo = 1;
    for (ind = 2; ind < numero; ind++){
        if ((numero % ind) == 0){
            ehPrimo = 0;
        }
    }

    if (ehPrimo){
        printf("O numero %d e primo.\n", numero);
    }
    else{
        printf("O numero %d nao e primo.\n",numero);
    }
}
```