

## O laço infinito

Uma estrutura de repetição pode apresentar execução infinita se:

- A condição de execução for mal formulada: a expressão realiza comparações exatas (`==` ou `!=`) ou envolve variáveis de controle cujo valor não se altera.
- A atualização da variável de controle não produz os valores esperados, devido às limitações na representação de números reais pelo computador.

```
float x, y;  
  
x = 1.0;  
y = 0.1; // atenção: este valor não admite representação exata!  
while( x != 0.0 )  
{  
    x = x - y;  
    printf("%.1f\n", x);  
}
```

## O operador ? : (ternário)

- É uma simplificação do comando if-else, ou seja, com apenas um comando e não blocos de comandos:
  - A forma geral do operador ? : é:
  - expressão condicional ? expressão1 : expressão2;
  - O funcionamento do operador ? é idêntico ao do comando if-else:  
primeiramente, a expressão condicional será avaliada e:
    - Se essa condição for verdadeira, o valor da expressão1 será o resultado da expressão condicional.
    - Se essa condição for falsa, o valor da expressão2 será o resultado da expressão condicional.
  - Ex: `int maior = (a>b ? a : b);`

## Comandos de controle de fluxo

Permitem alterar o fluxo de execução de uma estrutura de repetição.

`break;`

Ignora os comandos restantes do bloco, desviando imediatamente o fluxo de execução do programa para o primeiro comando após o final do bloco.

`continue;`

Ignora os comandos restantes do bloco e força a próxima iteração do laço, dependendo do valor da expressão de controle.

`exit(0);`

Encerra a execução do programa.

## ► Ex: break

### Comandos de controle de fluxo: break;

Exemplo: Adivinhe o número que estou pensando...

```
#include <time.h>

int main()
{
    int cont = 0, num, chute;

    srand( time(0) );           // inicializa o gerador de números aleatórios
    num = 1 + rand() % 100;     // num assume um valor inteiro entre 1 e 100;

    while ( 1 )                 // laço infinito
    {
        cont++;
        system("CLS");
        printf("Adivinhe o numero que estou pensando... [1, 100]: ");
        scanf("%d", &chute);
        if (chute == num)
            break;
    }
    printf("Acertou em %d tentativas\n", cont);
    system("PAUSE");
    return(0);
}
```

# Exercício 3 – Ling. C (Repetição)

---

1. Elabore um algoritmo em que o usuário entre com um número inteiro qualquer, e o software imprima os 20 números subsequentes ao que foi digitado pelo usuário.
2. Elabore um algoritmo que solicite que o usuário entre com dois números (inicial e final). Ao final o algoritmo deverá apresentar o valor total da soma de todos os números do intervalo digitado pelo usuário.
3. Elabore um algoritmo que apresente os números pares maiores que 10 no intervalo fechado [A, B]. Sendo que A e B serão números inteiros escolhidos pelo usuário. Um número é par quando este satisfaz a seguinte condição:  $(\text{NÚMERO} \bmod 2 = 0)$
4. Escreva um algoritmo que solicite que o usuário entre com valores inteiros quaisquer. Ao final imprima a quantidade de números digitados, o somatório dos valores digitados, e a média aritmética do somatório. O valor "0" (zero) indica parar de ler valores
5. Elabore um algoritmo para fazer cálculo de potenciação. Ou seja,  $x^y$ . (Exemplo:  $3^4 = 3 \times 3 \times 3 \times 3$ ). Seu algoritmo deverá solicitar que o usuário entre com o valor da base (x) e do expoente (y) e apresentar o resultado do cálculo sem utilizar os operadores `**` ou `^`. Para resolver o problema utilize estrutura de repetição.

# Exercício 3 – Ling. C (Repetição)

---

6. Suponha que seu computador consiga executar somente operações de soma e subtração. Escreva programas em C que, dados dois números inteiros positivos  $a$  e  $b$ , calculem:

- O produto  $a * b$ .
- O quociente e o resto da divisão de  $a$  por  $b$ .
- A potência  $a ^ b$  ( $ab$ ): Dica use o primeiro programa ( $a * b$ ) para fazer a potência.

FIM