

Aula 09 – Laços

Algoritmos e Lógica de Programação

Laços

Os laços são uma série de estruturas que permitem a execução de um código repetitivo de maneira controlada.

Eles formam a base da computação juntamente com as estruturas de testes e condições.

while

A instrução while executa um laço enquanto uma expressão lógica dá verdadeiro.

Ela respeita a seguinte sintaxe:

```
while(condição)  
    instrução;
```

while

Exemplo:

Faça um programa que conte de 0 até 10.

```
#include <stdio.h>
int main()
{
    int numero;
    numero = 0;
    while(numero <= 10)
    {
        printf("Numero: %d\n", numero);
        numero = numero + 1;
    }
    return 0;
}
```

while

Exemplo 2:

```
#include <stdio.h>
int main()
{
    char sn;
    sn = 'S';
    while(sn == 'S' || sn == 's')
    {
        printf("Deseja repetir esta mensagem?\n");
        scanf(" %c", &sn);
    }
    return 0;
}
```

for

A instrução for é extremamente poderosa e deve ser usada quando sabemos o número de iterações (repetições) que devemos ter.

Ela possui a seguinte sintaxe:

```
for(inicialização; condição; passo)  
    instrução;
```


for

Exemplo:

Faça um programa que conte de 0 a 10:


```
#include <stdio.h>
int main()
{
    int numero;
    for(numero = 0; numero <= 10; numero = numero + 1)
    {
        printf("Numero: %d\n", numero);
    }
    return 0;
}
```



for

Perceba que o for é uma estrutura muito mais compacta do que o while para iterações.

Um detalhe importante é que todas as partes do for são opcionais, ou seja, não existe a necessidade de termos inicialização, condição e passo. Logicamente que isto deve ser usado para auxiliar a lógica.



Operadores ++ e --

Existem dois operadores que facilitam muito o uso do for: ++ e --

Eles nada mais são que atalhos para estruturas do tipo `variável = variável + 1`.

Por exemplo:

`i++` é o mesmo que `i = i + 1`

`i--` é o mesmo que `i = i - 1`

for

Exemplo 2:

Faça um programa que conte de 0 a 10:

```
#include <stdio.h>
int main()
{
    int i;
    for(i = 0; i <= 10; i++)
    {
        printf("Numero: %d\n", i);
    }
    return 0;
}
```

do...while

A instrução do ... while é muito parecida com o while.

A diferença é que a condição é verificada no final, fazendo que o código interno seja executado pelo menos uma vez.

Sua sintaxe é:

```
do
```

```
    instrução;
```

```
while(condição);
```

do...while

Uma das maiores utilizações desta estrutura é forçar o usuário a digitar algo dentro de uma regra. Logicamente que ele pode ser usado para outras coisas.

Exemplo:

```
#include <stdio.h>
int main()
{
    char sn;
    do
    {
        printf("Deseja sair?");
        scanf(" %c", &sn);
    }
    while(sn != 'S' && sn != 's');
    return 0;
}
```

Exercícios

1. Faça um programa que coloque na tela a tabuada do 5 usando while.
2. Altere o programa anterior para aceitar a tabuada de qualquer número.
3. Faça um programa que imprima as tabuadas de 2 a 5 usando while.
4. Faça um programa que mostre o seguinte menu na tela e resolva as operações conforme o usuário seleciona-las:
 1. Soma
 2. Subtração
 3. Divisão
 4. Multiplicação
 5. Sair



Exercícios

5. Faça um programa que calcule a soma dos n primeiros números naturais.
6. Refaça os exercícios 1, 2 e 3 usando for
7. Faça um programa que recebe um número entre 1 e 10 inclusive e faça uma contagem regressiva deste número até zero.

Exercícios

8. Faça um programa que lê um valor ímpar inteiro n , ($2 < n < 20$) e imprima as figuras a seguir com n caracteres de lado (os exemplos mostram $n = 5$)

```
#####   ###   #           #   #####   #           #
#   #   #   #   ##           ##   #####   #           # #
#   #   #   #   ###           ###   ###   #####   #           #
#   #   #   #   #####           #####   ##   #           # #
#####   ###   #####           #####   #   #           #
```