

## Estrutura de Repetição "While "- (Enquanto )

- quando não se conhece de antemão o número de vezes que o conjunto de comandos no interior do laço será repetido, pelo fato do mesmo estar amarrado a uma condição sujeita à modificação pelas instruções do interior do laço.

### Linguagem C

```
#include <stdio.h>
#include <conio.h>
main()
{
    while(condição)
    {
        Comando(s)
    }
}
```

- O comando(s) é executado **enquanto** a condição for verdadeira .

Ex:

```
x=1; y=5
while (x < y)
{
    x=x+1;
    printf("teste");
}
```

### Programa em C

```
#include <stdio.h>
#include <conio.h>
main()
{
    float salario, soma_salario=0, media;
    int total=0;
    printf("digite o salario");
    scanf("%f",&salario) ;

    while(salario >0)
    {
        soma_salario=soma_salario+salario;
        total=total +1;
        printf("digite o salario  ");
        scanf("%f",&salario) ;
    }
    media=soma_salario/total;
    printf("A media dos salarios eh %.3f reais",media);
    getch();
}
```

## Estrutura de Repetição "do.....while " ( faça...enquanto.. )

Caracteriza-se por uma estrutura que efetua um teste lógico no final de um laço, verificando se é permitido ou não executar novamente o conjunto de comandos no interior do mesmo.

Seu funcionamento é bastante parecido ao da construção Enquanto. O comando é executado uma vez. A seguir, a condição é testada: se ela for verdadeira (diferente de zero), o comando composto é executado novamente e este processo é repetido até que a condição seja falsa(igual a zero), quando então a execução prossegue pelo comando imediatamente seguinte ao final da construção.

Esta construção difere da construção Enquanto pelo fato de o comando composto ser executado uma ou mais vezes (pelo menos uma vez), ao passo que na construção Enquanto o comando composto é executado zero ou mais vezes (possivelmente nenhuma). Isto acontece porque na construção **do...while** o teste é feito no final da construção, ao contrário do que acontece na construção **while**, onde o teste da condição é efetuado no início da mesma.

Forma Geral da Estrutura:

```
#include <stdio.h>
#include <conio.h>
main()
{
    do
    {
        Sequencia de comandos;
    }
    while (Expressão);
}
```

### Exemplos:

```
#include <stdio.h>
#include <conio.h>
main()
{
    float salario, soma_salario=0, media;
    int total=0;
    printf("digite o salario");
    scanf("%f",&salario) ;

    do
    {
        soma_salario=soma_salario+salario;
        total=total +1;
        printf("digite o salario  ");
        scanf("%f",&salario) ;
    }
    while(salario!=0);

    media=soma_salario/total;
    printf("A media dos salarios eh %.3f reais",media);
    getch();
}
```

## Comparação:

- Elabore um algoritmo, que leia 50 valores inteiros e calcule o produto dos valores lidos. Use o uma versão usando o comando FOR, outra o comando WHILE e outra usando o DO...WHILE

while	do...while	for
<pre> Cont=0; Prod=1; <b>While</b>( Cont &lt; 50) {     printf("digite um n° ");     Scanf ("%f",&amp;Num)     Prod=Prod * Num;     Cont=Cont+1; }     printf  ("Produto = %d ",Prod); </pre>	<pre> Cont=0; Prod=1; <b>do</b> {     printf("digite um n° ");     Scanf ("%f",&amp;Num);     Prod=Prod * Num;     Cont←Cont+1; } <b>While</b>( Cont&lt;50);  printf ("Produto=%d",Prod); </pre>	<pre> Prod=1; <b>for</b>(i=0;i&lt;50;i=i+1) {     printf("digite um n° ");     Scanf ("%f",&amp;Num);     Prod=Prod * Num; }     printf ("Produto=%d",Prod); </pre>

## Estruturas de Controle Encadeadas ou Aninhadas

- Um aninhamento ou encadeamento é o fato de se ter qualquer um dos tipos de construção apresentados anteriormente dentro do conjunto de comandos (comando composto) de uma outra construção.
- Em qualquer tipo de aninhamento é necessário que a construção interna esteja completamente embutida na construção externa.
- A figura a seguir ilustra aninhamentos válidos e inválidos.



(a)



(b)

Exemplos de aninhamentos (a) válidos e (b) inválidos