

C) ESTRUTURAS DE REPETIÇÃO

São muito comuns as situações em que se deseja repetir um determinado trecho de um programa um certo número de vezes.

As estruturas de repetição são muitas vezes chamadas de **Laços** ou também de **Loops**.

A classificação das estruturas de repetição é feito de acordo com o conhecimento prévio do número de vezes que o conjunto de comandos será executado. Assim os Laços se dividem em:

- **Laços Contados**, quando se conhece previamente quantas vezes o comando composto no interior da construção será executado;
- **Laços Condicionais**, 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.

Todo algoritmo que possui um ou mais de seus passos repetidos um determinado número de vezes denomina-se algoritmo com repetição.

Com a utilização de estruturas de repetição para a elaboração de algoritmos, torna-se necessário o uso de dois tipos de variáveis para a resolução de diversos tipos de problemas: variáveis contadoras e variáveis acumuladoras.

Uma variável **contador** é uma variável que recebe um valor inicial, geralmente 0 (zero) antes do início de uma estrutura de repetição, e é incrementada no interior da estrutura de um valor constante, geralmente 1, conforme o exemplo abaixo:

Uma variável **acumuladora** é uma variável que recebe um valor inicial, geralmente 0 (zero) antes do início de uma estrutura de repetição, e é incrementada no interior da estrutura de um valor variável, geralmente a variável usada na estrutura de controle, conforme o exemplo abaixo:

C.1) - Laços Contados

Os laços contados são úteis quando se conhece previamente o número exato de vezes que se deseja executar um determinado conjunto de comandos. Então, este tipo de laço nada mais é que uma estrutura dotada de mecanismos para contar o número de vezes que o corpo do laço (ou seja, o comando composto em seu interior) é executado.

Comando FOR

Forma Geral

```
for (inicializações; condições de manutenção da repetição; incrementos)
{
sequencia de comandos
}
```

Onde:

Inicializações — são atribuídos valores iniciais a variáveis;

condições de manutenção da repetição — estabelecem-se, através de uma expressão,

as condições nas quais a execução da sequência de comandos será repetida;

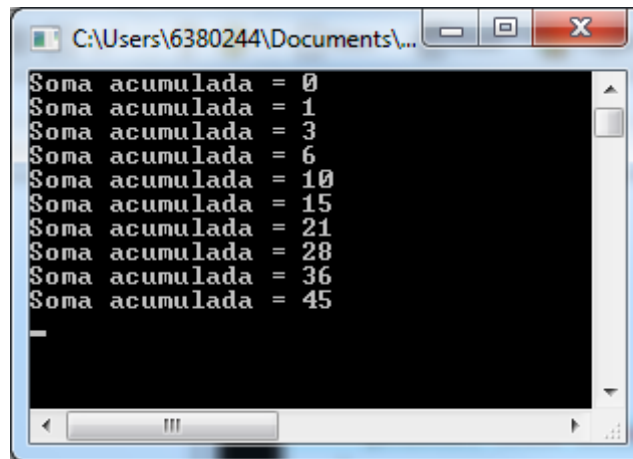
incrementos, — incrementam-se variáveis.

Quando um comando **for** é executado, a sequência de comandos da área das inicializações é executada. Em seguida, a expressão que fixa as condições de manutenção da repetição é avaliada. Se o valor desta expressão não for nulo, a sequência de comandos é executada, sendo em seguida executada a sequência de comandos da área dos incrementos. Novamente a expressão das condições de manutenção da repetição é avaliada e tudo se repete até que o seu valor seja igual a zero.

Exemplos:

Programa em C++

```
#include <stdio.h>
#include <conio.h>
main()
{
int soma=0,i;
for (i=0;i<=9;i=i+1)
{ soma=soma+i;
printf("Soma acumulada = %i \n", soma);
}
}
```



```
Soma acumulada = 0
Soma acumulada = 1
Soma acumulada = 3
Soma acumulada = 6
Soma acumulada = 10
Soma acumulada = 15
Soma acumulada = 21
Soma acumulada = 28
Soma acumulada = 36
Soma acumulada = 45
```

Programa em C ++

```
#include <stdio.h>
#include <conio.h>
main()
{ int i;
  for (i = 10; i >= 0; i = i - 2)
    printf("%d ", i);
}
```

exibe na tela os números 10, 8, 6, 4, 2, 0

```
#include <stdio.h>
#include <conio.h>
main()
{ int i;
  for (i = 1; i <= 10; i = i + 20)
    printf("%d ", i);
}
```

exibe, apenas, o numero 1

Fazer um programa em C++ para ler as notas de algoritmos de 20 alunos uma de cada vez . Verificar se a nota do aluno $\geq 7,0$ escrever “Aprovado” caso contrario escrever “Exame”. No final escrever a média aritmética das notas e escrever o número de alunos aprovados