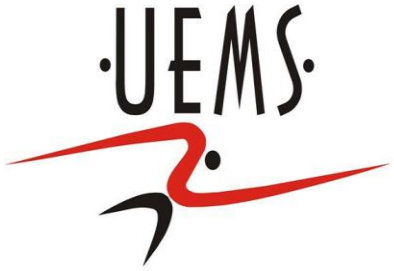


PCI- Estrutura de Repetição

Profa. Mercedes Gonzales Márquez

Estrutura de Repetição

- O computador muitas vezes é requerido para realizar tarefas *repetitivas* que seriam muito trabalhosas se realizadas de outra maneira.
- Em C, existem três tipos de estruturas de repetição, mas o princípio de funcionamento de todas é o mesmo: repetir um certo conjunto de comandos até que uma certa condição de parada seja satisfeita.

```
while (expressão) {
    sentença;
    sentença;
    ...
}
```

```
do {
    sentença;
    sentença;
    ...
} while (expressão);
```

```
for (inicialização;
     teste;
     atualização) {
   sentença;
   sentença;
   ...
}
```

- Esse tipo de estrutura costuma ser chamada de *laço* ou, equivalentemente em inglês, *loop*.

Estrutura de Repetição

- Exemplo: Imprima uma sequência de números na tela, especificamente entre 1 e 5

  A primeira ideia seria usar a estrutura sequencial e fazermos algo como:

  printf (“1”);

  printf (“2”);

  printf (“3”);

  printf (“4”);

  printf (“5”);

  Isto não é prático, mas ainda se considerarmos uma sequência muito grande como imprimir os números entre 1 e 1.000.000.

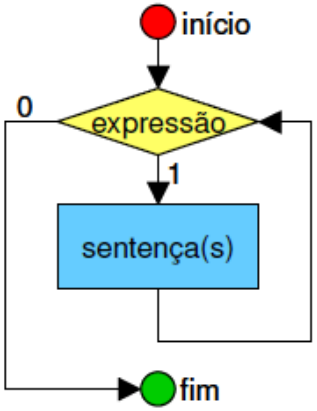
Estrutura de Repetição

- Nestes casos usamos uma variável para guardar um primeiro número inteiro e depois incrementamos repetidamente esse número. Vejamos a solução usando estrutura *while*:

```
i=1;
while (i<=1000000){ /*enquanto a variável i não atingir
    printf("%d",i);          o valor 1000000, ela é impresa
    i++;                     e incrementa-se seu valor em 1 */
}
```

Estrutura de Repetição

- No tipo de laço **while** (*enquanto*) uma condição é avaliada; se ela for falsa, continua a execução do resto do programa; se ela for verdadeira, um bloco de comandos é executado, e volta-se ao começo do processo (avaliando a condição novamente).
- Em outras palavras, o bloco de comandos é executado *enquanto a condição for satisfeita*.




```
while (expressão) {
    sentença;
    sentença;
    ...
}
```

```
while (expressão)
    sentença;
```

Estrutura de Repetição

- Cada execução do bloco de comandos é chamada de **iteração** — não só para o while, mas também para os outros laços que veremos posteriormente.
- Temos duas observação importantes a fazer:

1. Se a condição já for falsa quando o while for atingido, nenhuma iteração será executada e o bloco de comandos será ignorado. Ex:

```
while (a!=a)
    a=a+1;
```

Nunca entra na repetição (loop).

2. *Se a condição for* sempre verdadeira, entra-se na repetição e nunca se sai (*loop infinito).* Ex:

```
while (a == a)
    a=a+1;
```

Estrutura de Repetição

Dica:

- Para ajudar a implementar sua estrutura de repetição, você pode pensar na condição de repetição ou alternativamente, na condição de parada.
- Tenha em conta que uma é a negação da outra.

Condição de repetição = ! Condição de parada
Ou
Condição de parada = ! Condição de repetição

- Tenha em conta que será a condição de repetição que deverá ir do lado do comando while.

Estrutura de Repetição

Exemplo 1: Imprimir os 100 primeiros números inteiros

```
#include <stdio.h>
int main() {
   int i=1;
   while (i<=100){
      printf("%d",i);
      i++;
   }
}
```

```
#include <stdio.h>
int main() {
   int i=1;
   while (i<=100)
      printf("%d",i++);
}
```

Observe que será impresso o valor da variável i e os subsequentes enquanto i for <= 100.

A condição de parada é !(i<=100) =i>100.

Estrutura de Repetição

Exemplo 2: Imprimir os n primeiros números inteiros

```
#include <stdio.h>
int main() {
   int i=1, n;
   printf(“Informe um numero:”);
   scanf (“%d”,&n);
   while (i<=n){
        printf(“%d”,i);
        i++;
   }
}
```

```
#include <stdio.h>
int main() {
   int i=1, n;
   printf(“Informe um numero:”);
   scanf (“%d”,&n);
   while (i<=n)
        printf(“%d”,i++);
}
```

A condição de parada é !(i<=n) =i>n.

Estrutura de Repetição

- Os exemplos 1 e 2 embutem um conceito muito importante em programação conhecido como **contador** — uma variável usada em um laço, cujo valor varia em uma unidade a cada iteração.
- Esse padrão é usado em diversas situações que envolvam a repetição de várias operações iguais ou bem parecidas, que possam ser diferenciadas apenas pelo valo do contador.

Estrutura de Repetição

Exemplo 3: Faça um programa que imprima S=1+2+3+...+(n-2)+(n-1)+n, para um n fornecido pelo usuário.

- Usamos uma técnica elementar em programação que é o uso de **acumuladores**.
- O acumulador usado é uma variável cujo papel é incorporar ou acumular valores.
- Somas ou produtos acumulativos são frequentes em cálculos de somatórias, produtórias ou consolidações de resultados (totais, médias, ...).

Estrutura de Repetição

Somatórios e produtórios

- Os métodos estatísticos frequentemente se utilizam das técnicas de somatório (Σ (sigma)) e produtório (Π (pi)).

$$\sum_{i=1}^{n} x_i = x_1 + x_2 + x_3 + \ldots + x_n;$$

$$\prod_{i=1}^{n} x_i = x_1 \cdot x_2 \cdot x_3 \cdot \ldots \cdot x_n \ ;$$

x	representa a variável
i	é dito o indexador (índice)
n	é o valor final do indexador,

Estrutura de Repetição

Somatório e produtório

- As vezes o indexador não inicia em 1.

$$\sum_{i}^{n} x_i = x_k + x_{k+1} + \ldots + x_{K+n-1} + x_{k+n}$$

$$\prod_{l=k}^{n} x_l = x_k \times x_{k+1} \times \ldots \times x_{k+n-1} \times x_{k+n}$$

x	representa a variável
i	é dito o indexador (índice)
k	é o valor inicial do indexador, sendo que o caso especial é quando $k = 1$
n	é o valor final do indexador,

Estrutura de Repetição

Exemplo de somatórios:

- Soma da variável índice

$$\sum_{i=1}^{n} i = 1 + 2 + 3 + \cdots + n$$

- Soma simples

$$\sum_{i=1}^{n} x_i = x_1 + x_2 + \ldots + x_n$$

- Soma de quadrados

$$\sum_{i=1}^{n} x_i^2 = x_1^2 + x_2^2 + \ldots + x_n^2$$

- Quadrado da soma

$$\left(\sum_{i=1}^{n} x_i \right)^2 = (x_1 + x_2 + \ldots + x_n)^2$$

- Soma de produtos

$$\sum_{i=1}^{n} x_i y_i = x_1 y_1 + x_2 y_2 + \ldots + x_n y_n$$

Estrutura de Repetição

Exemplo de somatórios :

$$X = \{15, 23, 22, 40\}$$
$$Y = \{24, 53, 30, 31\}$$

a. $\sum_{i=1}^{4} x_i = 15 + 23 + 22 + 40 = 100$

b. $\sum_{i=1}^{4} x_i^2 = 15^2 + 23^2 + 22^2 + 40^2 = 2838$

c. $\left(\sum_{i=1}^{4} x_i\right)^2 = (15 + 23 + 22 + 40)^2 = 10000$

d. $\sum_{i=1}^{4} x_i y_i = 15 \times 24 + 23 \times 53 + 22 \times 30 + 40 \times 31 = 3479$

e. $\left(\sum_{i=1}^{4} x_i\right)\left(\sum_{i=1}^{6} y_i\right) = 13800$

Estrutura de Repetição

Exemplo de produtórios:

- Produto simples

$$\prod_{i=1}^{n} x_i = x_1 \times x_2 \times ... \times x_n$$

- Produto de uma constante k

$$\prod_{i=1}^{n} k = k \times k \times ... \times k = k^n$$

- Produto de uma constante por uma variável

$$\prod_{i=1}^{n} kx_i = kx_1 \times kx_2 \times ... \times kx_n = k^n \prod_{i=1}^{n} x_i$$

- Produto de duas variáveis

$$\prod_{i=1}^{n} x_i y_i = x_1 y_1 \times x_2 y_2 \times ... \times x_n y_n = \prod_{i=1}^{n} x_i \prod_{i=1}^{n} y_i$$

- Produto de uma variável índice

$$\prod_{i=1}^{n} i = 1 \times 2 \times 3 \times ... \times n = n! \ \ (leia - se: \ n \ fatorial)$$

Estrutura de Repetição

Exemplo de produtórios :

Considere amostras de tamanho $n = 3$ das variáveis X e Y:

$$X = \{1, 3, 4\}$$
$$Y = \{2, 5, 0\}$$

a. $\prod_{i=1}^{3} x_i = 1 \times 3 \times 4 = 12$

b. $\prod_{i=1}^{3} y_i = 2 \times 5 \times 0 = 0$

c. $\prod_{i=1}^{3} 2x_i = 2^3 \prod_{i=1}^{3} x_i = 8 \times 12 = 96$

d. $\prod_{i=1}^{3} x_i y_i = \prod_{i=1}^{3} x_i \prod_{i=1}^{3} y_i = 12 \times 0 = 0$

Estrutura de Repetição

Exemplo 3: Faça um programa que imprima S=1+2+3+...+(n-2)+(n-1)+n, para um n fornecido pelo usuário.

- Precisamos acumular as somas parciais a cada número lido. Utilizaremos para isso uma variável chamada *soma*, que deverá ser inicializada com o valor zero, e à qual se somará cada número lido.
- A inicialização do acumulador é com um valor base que neste caso será o valor zero (elemento neutro da adição). Se fosse uma multiplicação dos números a partir do 1, usaríamos como valor base do produto o elemento neutro da multiplicação, isto é, o 1.

Estrutura de Repetição

- Soma da variável índice $\sum_{i=1}^{n} i = 1 + 2 + 3 + \cdots + n$

```
int i=1,n, soma= 0;
printf(“Digite o número n : ");
scanf("%d", &n);
while (i<=n) {
    soma = soma + i;
    i++;
}
printf("Soma = %d\n", soma);
```

Valor inicial=
Neutro da soma

Condição de parada
com contador

Acumulador

Contador

Podemos também simplificar o while com um único comando, assim:

```
while (i<=n)
    soma += i++;
```

Estrutura de Repetição

- Exemplo 4: Faça um programa que imprima P=1x2x3x...x(n-1)xn, para um n fornecido pelo usuário. Veja que se trata do produto da variável índice ou fatorial de n. $\prod_{i=1}^{n} i = 1 \times 2 \times 3 \times ... \times n = n!$

```
int i=1, n, prod = 1;
printf("Digite o número n : ");
scanf("%d", &n);
while (i<=n)
    prod *= i++; /*produto=produto+i*/
printf("Fatorial = %d\n", prod);
```

Estrutura de Repetição

Exemplo 5: Imprima o fatorial de um número inteiro n com a fórmula n!= n x (n -1) x (n - 2) x ... x 3 x 2 x1. Pela comutatividade da multiplicação, esta fórmula com a do exemplo 4 são equivalentes, mesmo que os valores estejam em ordem decrescente.

```
int i,n, prod = 1;
printf("Digite o número n : ");
scanf("%d", &n);
i=n;
while (i>0)
    prod *= i--; /*produto=produto-i*/
printf("Fatorial = %d\n", prod);
```

Estrutura de Repetição

Exemplo 6: Imprima a média das idades dos alunos da turma de Sistemas de Informação do 1º ano.

$$\bar{X} = \frac{x_1 + x_2 + \cdots + x_n}{n} = \frac{\sum_{i=1}^{n} x_i}{n}$$

```
int i=1,n, idade;
float media;
printf("Digite o número n de alunos de SI : ");
scanf("%d", &n);
while (i<=n){
   printf (“Informe a idade:”);
   scanf (“%d”,&idade);
   media += idade; /*media=media+idade*/
   i++;
}
printf(“Media = %2.1f\n", media/n);
```

Estrutura de Repetição

Exemplo 7: Imprima o elemento máximo da sequência de números inteiros positivos digitada pelo usuário.

- Como você realizaria essa tarefa? Imagine alguém ditando para você tal sequência de números e você com o pincel e o quadro branco.
- Você anota o primeiro número e, caso ouça posteriormente um número maior que ele, anota-o logo abaixo; se ouvir outro número ainda maior, anota-o abaixo do anterior; e assim por diante.
- Não é necessário tomar nota de todos os números. No final, o último número anotado será o maior número da sequência.

Estrutura de Repetição

Exemplo 7: Imprima o elemento máximo da sequência de números positivos digitada pelo usuário.

- Um computador poderia fazer isso exatamente da mesma maneira. Em vez de ‘anotar’ os números, ele os guardaria em variáveis; além disso, ele não precisaria manter as ‘anotações’ anteriores; basta substituir o valor da variável.

Estrutura de Repetição

Exemplo 7: Imprima o elemento maximo de uma sequência de números positivos digitada pelo usuário.

```
int i=1, n, num, max; /* candidato a máximo */
printf(“Digite o número n de elementos : “);
scanf("%d", &n);
max = 0;
while (i<=n){
   printf (“Informe o numero:”);
   scanf (“%d”,&num);
   if (num > max) /* para procurar o máximo */
      max = num;
   i++;
}
printf("Máximo = %d\n", max);
```

Estrutura de Repetição

Exemplo 8: Agora imagine que você não conhece apriori o número de inteiros positivos da sua sequência do exemplo 7. Então a condição de repetição não tenha um limite pré-definido.

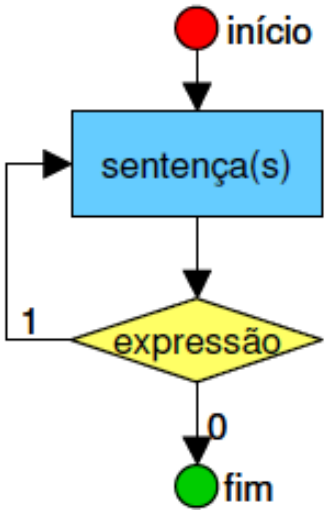
- Como se trata de um conjunto de números positivos, nós podemos estabelecer como condição de parada que o número informado seja negativo ou zero.

Estrutura de Repetição

```
int num, max; /* candidato a máximo */
printf(“Informe o primeiro numero n da sequencia: “);
scanf("%d", &num);
max = 0;
while (num>0){
   if (num > max) /* para procurar o máximo */
      max = num;
  printf (“Informe o proximo numero:”);
  scanf (“%d”,&num);
}
printf("Máximo = %d\n", max);
```

Estrutura de Repetição

- O tipo de laço **do** ... **while** tem um comportamento muito semelhante ao **while**, com uma diferença crucial: a condição é verificada após executar o bloco de instruções correspondente.




```
do {
    sentença;
    sentença;
    ...
} while (expressão);
```

```
do sentença while (expressão);
```

Estrutura de Repetição

Exemplo 1: Imprimir os n primeiros números inteiros positivos.

```
int i, n;
i=1;
scanf("%d",&n);
do{
    printf("\n %d",i);
    i++;
}while(i<=n);
```

```
int i, n;
i=1;
scanf("%d",&n);
do
    printf("\n %d",i++);
while(i<=n);
```

O que acontece se o usuário digitar 0 (n=0)?
O que acontece se usarmos o while?

Estrutura de Repetição

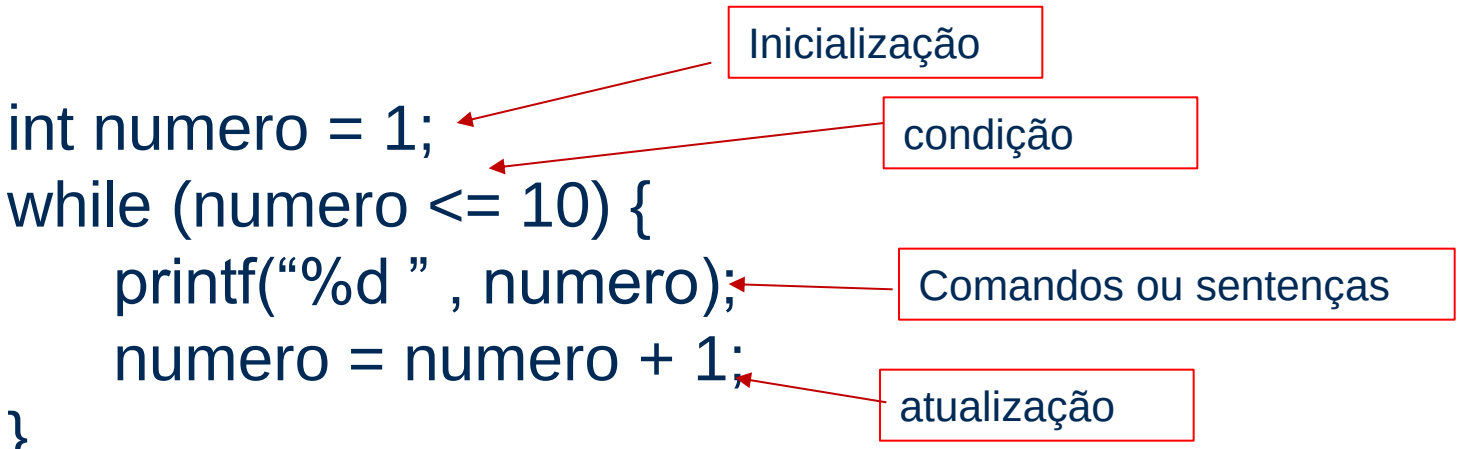
Exemplo 2: Faça um algoritmo que determine os quadrados de um conjunto de números inteiros positivos.

```
int num;
scanf (“%d”,&num);
do{
    printf(“%d”,num*num);
    scanf (“%d”,&num);
}while (num>0);
```

O que acontece se o usuário digitar 0 (num<=0)?
O que acontece se usarmos o while?

Estrutura de Repetição

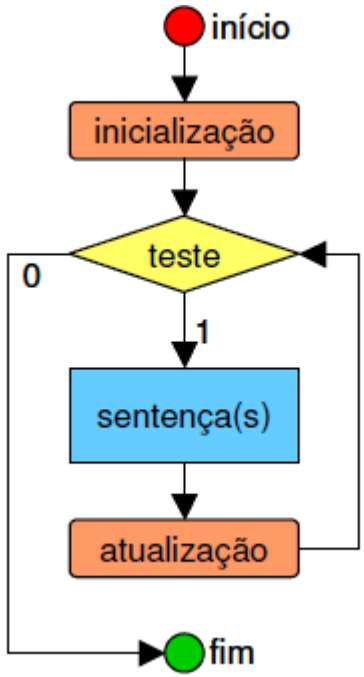
Vejamos um programa com while com suas 4 partes fundamentais:




```
int numero = 1;
while (numero <= 10) {
      printf(“%d ” , numero);
      numero = numero + 1;
}
```

Estrutura de Repetição

Estrutura for: Incorpora como parâmetros a inicialização, o teste condicional e a atualização.




```
inicio;
for (inicialização;
   teste;
   atualização) {
      sentença;
      sentença;
      ...
}
fim;
```

```
int numero;
for (numero = 1; numero <= 10; numero++) {
      printf("%d ", numero);
}
```