

Aula 10 – Comandos de Repetição For e Do While

1) O comando for

É muito comum usarmos o comando **while** para repetir alguns comandos um número definido de vezes. Fazemos isso usando um contador.

Considere os problemas abaixo:

P13) Dado $n > 0$ e uma sequência de n números calcular a soma dos elementos da sequência:

```
#include <stdio.h>
#include <stdlib.h>
/* dado n>=0 e uma sequencia de n numeros, calcular a soma dos números.*/
int main() {
    int n, /* numero de elementos */
        cont, /* contador */
        x, /* numero lido */
        soma; /* soma dos elementos */

    /* ler o n */
    printf("digite o valor de n:");
    scanf("%d", &n);
    cont = 1; /* inicie o contador */
    soma = 0; /* inicia a soma com zero */
    /* repita n vezes a leitura e soma */
    while (cont <= n) {
        printf("\ndigite um elemento da sequencia:");
        scanf("%d", &x);
        soma = soma + x;
        cont = cont + 1; /* incrementa o contador */
    }
    /* imprima o resultado */
    printf("\nsoma dos elementos da sequencia:%10d", soma);
    system("pause");return 0;
}
```

P19) Dado $n > 0$ calcular os divisores de n

```
#include <stdio.h>
#include <stdlib.h>
/* dado n>=0 calcular os divisores proprios de n (1<d<n)*/
main() {
    int n, /* numero de elementos */
        d; /* divisores */
    /* ler o n */
    printf("digite o valor de n:");
    scanf("%d", &n);
    d = 2; /* 2 e o primeiro candidato a divisor */
    /* repita n/2 vezes testando todos os possiveis divisores */
    while (d <= n/2) {
        if (n%d == 0) printf("\n%10d divisor de %10d", d, n);
        d = d + 1; /*incrementa o contador ou o novo candidato a divisor*/
    }
    system("pause");return 0;
}
```

A construção abaixo, é muito comum:

```
i = 1; /* inicialização de contador */  
while (i <= n) /* comparação - para ou continua a repetir? */  
{  
  :  
  :  
  :  
  i = i + 1; /* incremento de contador */  
}
```

O comando **for** faz exatamente isso de forma abreviada.

```
for (i = 1; i <= n; i = i + 1)  
{  
  :  
  :  
  :  
}
```

De uma forma geral, a construção abaixo, é muito comum:

```
<inicialização>; /* inicialização de contador */  
while (<comparação>) /* comparação - para ou continua a repetir? */  
{  
  :  
  :  
  :  
  <incremento>; /* incremento de contador */  
}
```

O comando for abaixo faz exatamente isso de forma abreviada:

```
for (<inicialização>; <comparação>; <incremento>)  
{  
  :  
  :  
  :  
}
```

Na verdade o comando for é mais geral que isso, como será visto mais tarde. Por ora, é bom entendê-lo como um caso particular do while. Observe que os elementos comuns (<inicialização>, <comparação> e <incremento>) foram colocados dentro do comando for.

Desta forma os programas acima ficariam:

P13)

```
#include <stdio.h>  
#include <stdlib.h>  
/* dado n>=0 e uma sequencia de n numeros, calcular a soma dos números */  
int main() {  
  int n, /* numero de elementos */  
      cont, /* contador */  
      x, /* numero lido */  
      soma; /* soma dos elementos */  
  
  /* ler o n */
```

```
printf("digite o valor de n:");
scanf("%d", &n);
/* repita n vezes a leitura e soma */
soma = 0; /* inicia a soma com zero */
for (cont = 1; cont <= n; cont = cont + 1) {
    printf("\ndigite um elemento da sequencia:");
    scanf("%d", &x);
    soma = soma + x;
}
/* imprima o resultado */
printf("\nsoma dos elementos da sequencia:%10d", soma);
system("pause");return 0;
}
```

O comando **for** no exemplo acima pode ser escrito de outras formas:

```
for (cont = 1; cont <= n; cont++)
```

ou

```
for (cont = n; cont >=1; cont = cont - 1)
```

ou

```
for (cont = n; cont >=1; cont--)
```

P19)

```
#include <stdio.h>
#include <stdio.h>
/* dado n>=0 calcular os divisores proprios de n (1<d<n)*/
int main() {
    int n, /* numero de elementos */
        d; /* divisores */
    /* ler o n */
    printf("digite o valor de n:");

    scanf("%d", &n);
    /* repita para d = 2 ate n/2 testando todos os possiveis divisores */
    for (d = 2; d <= n/2; d = d + 1)
        if (n%d == 0) printf("\n***%10d divisor de ***%10d", d, n);
    system("pause");return 0;
}
```

O comando **for** no exemplo acima pode ser escrito de outras formas:

```
for (d = 2; d <= n/2; d++)
```

ou

```
for (d = n/2; d >= 2; d = d - 1)
```

ou

```
for (d = n/2; d >= 2; d--)
```

P30) Dado $n \geq 0$ calcular $n!$

Como sabemos: $n! = n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot 2 \cdot 1$ se $n > 0$ e $n! = 1$ se $n=0$.

```
#include <stdio.h>
#include <stdlib.h>
/* dado n>=0 calcular o fatorial de n */
int main() {
    int n, /* numero dado */
        fat, /* fatorial */
        i; /* contador */
    /* ler o n */
    printf("digite o valor de n:");
    scanf("%d", &n);
    fat = 1; /* inicia o valor de fat */
    /* multiplique por todos os números até n */
    for (i = 1; i <= n; i = i + 1) fat = fat * i;

    /* imprima o resultado */
    printf("\nfatorial de %10d - %10d", n, fat);
    system("pause");return 0;
}
```

O comando `for` no exemplo acima pode ser escrito de outras formas:

```
for (i = 1; i <= n; i++)
```

ou

```
for (i = N; i >= 1; i = i - 1)
```

ou

```
for (i = N; i >= 1; i--)
```

P31a) Calcular o valor da função $x^3 + x^2 + x + 1$ para $x = -10, -9, \dots, 9, 10$

P31) Dado um intervalo de inteiros $[a, b]$, calcular o valor da função $x^3 - x^2 + x - 1$ para $a \leq x \leq b$

```
#include <stdio.h>
#include <stdlib.h>
c dado um intervalo de inteiros [a, b] onde a < b */
/* calcular o valor da funcao para x = a, a+1, ... , b */
int main() {
    int a, b, /* intervalo dado */
        f, /* valor da funcao calculada */
        i; /* contador */
    /* ler a e b */
    printf("digite o valor de a:");
    scanf("%d", &a);
    printf("\ndigite o valor de b:");
    scanf("%d", &b);
    /* calcule o valor da funcao para todos os inteiros entre a e b */
    for (i = a; i <= b; i = i + 1) {
```

```
    f = i*i*i - i*i + i - 1;
    /* imprima o resultado */
    printf("\nvalor da funcao para x= %10d - %10d", i, f);
}
system("pause");return 0;
}
```

P32) Dado $n \geq 0$ e x diferente de zero, inteiros, calcular x^n (x elevado a n). Sem usar a função **pow** do C.

```
#include <stdio.h>
#include <stdlib.h>
/* dado n>=0 e x diferente de zero calcular x elevado a n */
int main() {
    int n, /* expoente */
        x, /* base */
        pot, /* resultado */
        i; /* contador */
    /* ler o n */
    printf("digite o valor de n:");
    scanf("%d", &n);
    /* ler o x */
    printf("\ndigite o valor de x:");
    scanf("%d", &x);

    pot = 1; /* inicia o valor de fat */
    /* multiplique por x n vezes */
    for (i = 1; i <= n; i = i + 1) pot = pot * x;

    /* imprima o resultado */
    printf("\n%10d elevado a %10d resulta %10d", x, n, pot);
    system("pause"); return 0;
}
```

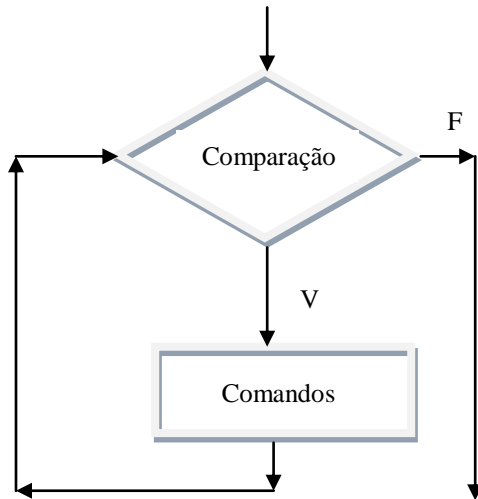
P32a) Dado $n \geq 0$ e x diferente de zero, inteiros, calcular $x^n + x^{n-1} + x^{n-2} + \dots + x^2 + x + 1$.
Sugestão: note que cada termo da soma, é o anterior multiplicado por x .

Agora como exercício, resolva todos os exercícios que já resolvidos com **while**, usando o comando **for**.

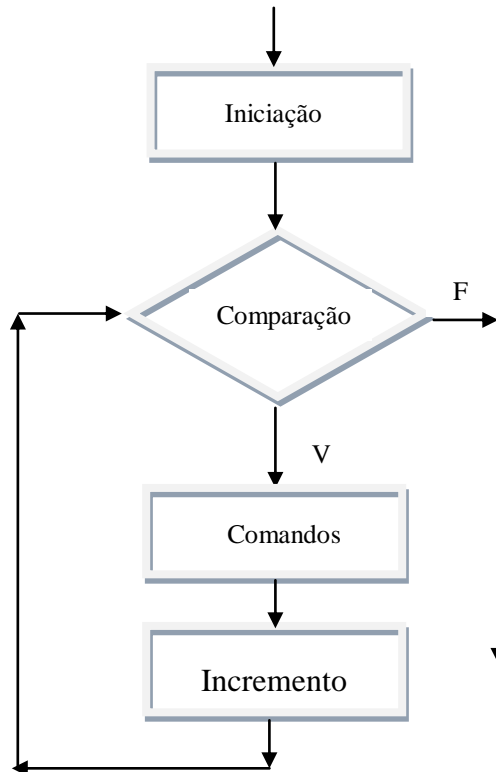
Fluxograma dos comandos while e for

Compare agora o fluxograma do comando while e do comando for.

while



for



2) O comando do while

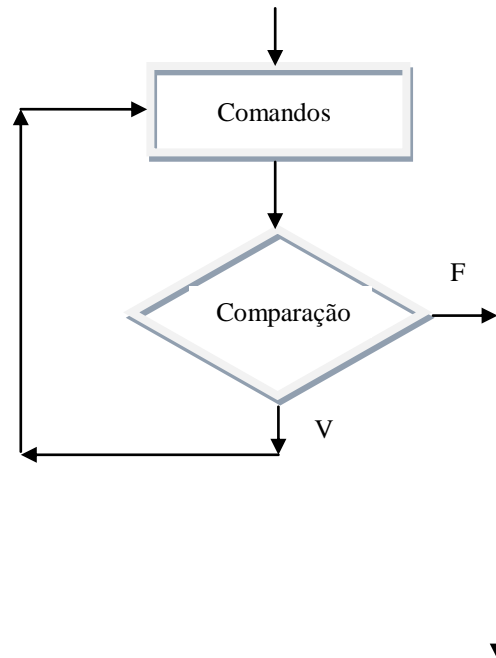
É um novo comando de repetição parecido com o comando while. A diferença é que a comparação é feita após a repetição. A repetição continua enquanto a comparação for verdadeira.

Formato geral:

```
do {c1; c2; ...; cn}  
while (comparação)
```

c1, c2, ..., cn, são os comando a serem repetidos.
Quando houver só c1 não é necessário colocar as chaves.

Veja abaixo o fluxograma deste comando e compare com o do comando while.



Vejam agora alguns exemplos usando esse comando.

P13) Dada uma seqüência de números terminada por zero, calcular a soma dos elementos da seqüência.

```
#include <stdio.h>
#include <stdlib.h>
/* dada uma sequencia de números terminada por zero, calcular a soma.
int main() {
    int x, /* numero lido */
        soma; /* soma dos elementos */

    soma = 0; /* inicia a soma com zero */

    /* repita ler e somar ate encontrar o fim da sequencia */
    do {printf("\ndigite um elemento da sequencia:");
        scanf("%d", &x);
        soma = soma + x;
    }
    while (x != 0);

    /* imprima o resultado */
    printf("\nsoma dos elementos da sequencia:%10d", soma);
    system("pause"); return 0;
}
```

Tente também resolver os que já resolvemos com o **while**, usando o **do while**.

Importante – o **do while** repete pelo menos uma vez. Isso não acontece com o **while** ou o **for** que podem não repetir nenhuma vez, no caso que da comparação ser falsa logo na primeira vez.

A repetição definida – Repetir um conjunto de comandos um determinado número de vezes

A repetição de um número determinado de vezes pode ser feita com qualquer um dos comandos: **while**, **for** ou **do while**.

Veja abaixo a equivalência:

```
i = 1;
while (i <= n)
  {comandos;
   i = i + 1;
  }
```

```
for (i = 1; i <= n; i = i +1)
  {comandos;}
```

```
i = 1;
do {comandos;
   i = i + 1;
  }
while (i <= n)
```

Se $n > 0$, são todas equivalentes. Quando $n = 0$, as duas primeiras não executam **comandos**. A terceira executa **comandos** uma vez.