

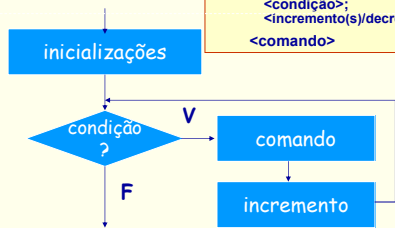
Comando de repetição – número fixo de vezes

C

for (expressão 1; expressão 2; expressão 3)
<comando>;

Linguagem algorítmica

para (<inicialização(s)>;
<condição>;
<incremento(s)/decremento(s)>)
<comando>



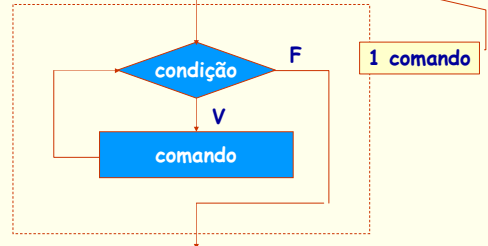
Comando while

C

Linguagem algorítmica

while (condição)
<comando>

enquanto (condição)
<comando>



Comandos iterativos (loop)

- *para tantas vezes repita ...*
- *enquanto*
- *faça ... enquanto* 

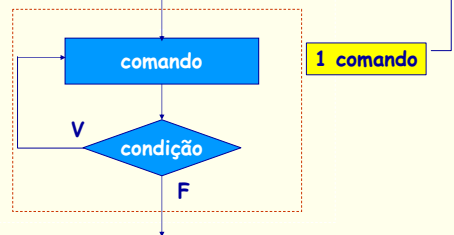
Comando do...while

C

Linguagem algorítmica

do
comando;
while (condição);

faça
comando
enquanto (condição)



Ex: Processar as notas dos alunos de uma turma (não sabe quantos serão!).

Para cada aluno, ler suas 3 notas, calcular e informar a média.
Para #include <stdio.h>
No: #include <math.h> // para utilizar round – fundamental em divisões reais

```

int main ( )
{
    float n1, n2, n3; //notas de um aluno
    float media, soma=0; // média e acumulador de médias da turma
    int al = 0; //variável de contar alunos
    system("color f1"); // fundo branco, letra azul
    do
    {
        printf("Informe notas do aluno %d: ", al + 1);
        scanf("%f", &n1); // lê só a primeira nota do 1º aluno
        if (n1 <= 100)
        { // só executa se a 1ª nota for <= 100
            al = al + 1;
            scanf("%f", &n2); // lê a segunda nota do aluno
            media = (n1 + n2 + n3)/3.0;
            soma = soma + media;
            printf("Média do aluno %d : %.2f", al, media);
        }
    } while (n1 <= 100);
    printf("Média da turma: %.2f", soma/float(al));
    system("pause");
    return 0;
}
  
```

testa sempre 2 vezes!

Ex: Processar as notas dos alunos de uma turma (não sabe quantos serão!).

Para cada aluno, ler suas 3 notas, calcular e informar a média.
Para encerrar a turma, informar a primeira nota com valor superior a 100
Para #include <stdio.h>
No: #include <math.h> // para utilizar round – fundamental em divisões reais

```

int main ( )
{
    float n1, n2, n3; //notas de um aluno
    float media, soma=0; // média e acumulador de médias da turma
    int al = 0; //variável de contar alunos
    system("color f1"); // fundo branco, letra azul
    printf("Informe notas do aluno %d: ", al + 1);
    scanf("%f", &n1); // lê só a primeira nota do 1º aluno
    if (n1 <= 100)
    {
        // só executa se a 1ª nota for <= 100
        al = al + 1;
        scanf("%f", &n2); // lê a segunda nota do aluno
        media = (n1 + n2 + n3)/3.0;
        soma = soma + media;
        printf("Média do aluno %d : %.2f", al, media);
        printf("Informe notas do aluno %d: ", al + 1);
        scanf("%f", &n1); // lê só a primeira nota do PRÓXIMO aluno
    }
    while (n1 <= 100);
    printf("Média da turma: %.2f", soma/float(al));
    system("pause");
    return 0;
}
  
```

```
#Calcula media de um numero indeterminado de valores lidos
#include <stdlib.h> // para usar system("PAUSE") no final
#include <stdio.h>
int main ( )
{
    int numero, lidos = 0;
    long int soma = 0;
    printf("Digite um numero inteiro de cada vez.");
    printf("\nPara terminar, digite o valor 9999.");
    do // sempre executa 1 vez?
    {
        printf("\nNumero? ");
        scanf("%d", &numero);
        if (numero != 9999)
        {
            soma = soma + numero;
            lidos = lidos + 1;
        }
    } while (numero != 9999);
    printf("\nMedia dos valores lidos: %.5f\n", (float)soma/lidos);
    system("PAUSE"); // segura a tela de execucao
    return 0;
}
```

```
// solução com while – neste caso, melhor!
printf("\nNumero? "); scanf("%d", &numero);
while (numero != 9999)
{
    soma = soma + numero;
    lidos = lidos + 1;
    printf("\nNumero? "); scanf("%d", &numero);
}
```

```
...
a = 0;
do
{
    a = a + 1;
    printf("%d", a);
} while (a < 5);
...
```

```
...
achou = 0;           // 0 = falso
do
{
    scanf("%d",&numero);
    printf("%d", numero);
    if (numero == 250)
        achou = 1; // equivale a true
} while ( ! achou); // pára se achou = 1
```

```
for (a = 1; a<=5; a++)  
    printf("%d", a);
```

```

...
a = 1;
do
{
    printf("%d", a);
    a = a + 1;
} while (a <= 5);
printf("%d", a);
...

```

```
do
{
    scanf("%d", &numero);
    printf("%d", numero);
    printf("Outro ? (1 = sim / 0 = nao) ");
    scanf("%d", &resposta)
}
while (resposta == 1);
```

```
#Programa Peso_ideal, utilizando o do-while como recurso para consistência na
sexo informado: ?/
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
int main()
{
    float altura,pesoideal;
    char sexo;
do
    {
        // laço para só ir adiante quando sexo informado for correto:
        printf("digite o sexo M/F\n");
        scanf("%c", &sexo); // branco antes do % faz com que ignore retorno de carro anterior
        sexo = toupper(sexo);
        printf("%c",sexo);
    } while (sexo != 'M' && sexo != 'F');

    printf("digite a altura\n");
    scanf("%f",&altura);
    if (sexo=='M')
        pesoideal = altura*72.7 - 58;
    else
        pesoideal = altura*62.1 - 44.7;
    printf("seu peso ideal e %f\n",pesoideal);
    printf("o sexo eh %c\n",sexo);
    system("pause");
    return 0;
}
```

comando	verifica condição	repetições	execução do(s) comando(s)
for	início	0 ou +	condição = V
while	início	0 ou + (?)	condição = V
do-while	final	<u>1</u> ou + (?)	condição = V

Programação Estruturada - observações importantes:

- ✓ Comandos de repetição só devem ser interrompidos através de modificações:
 1. detectadas a partir do cabeçalho do for (modificações especificadas para a variável de controle)
 - ou
 2. por modificações que se reflitam nas condições das expressões de controle do while e do-while.
- ✓ O uso de break como mecanismo de interrupção de uma repetição é totalmente vetado!!!
- ✓ A(s) variável(is) de controle do for só deve(m) ser alterada(s) pela(s) operação(ões) definida no cabeçalho do comando!

```
/descreve 20 vezes Bom dia
#include <stdio.h>
#include <stdlib.h>
int main ( )
{
    int cont,i;
    system("color f1");
    cont = 0;
    do
    {
        for (i=1;i<=cont;i++)
            printf(" ");
        printf("Bom Dia !!!\n");
        cont = cont + 1;
    }
    while (cont < 20);
    system("pause");
    system("cls");
    return 0;
}
```

[illegible]

1 - Complete o programa abaixo, de forma que os números pares entre 10 e 100 sejam gerados, somados e a que soma obtida seja informada na tela. Para isto, utilize o comando while, o comando do_while e o comando for. A tela de execução do programa deve ser como mostrada abaixo.

```
/* soma pares e mostra equivalências entre for, while e do while: */
#include <stdio.h>
#include <stdlib.h>
int main ( )
{
    system("color f1"); // muda cor neste comando!!!
    int par, soma;
    // usando while:

    // usando do while:

    // usando for:

    system("pause");
    return 0;
}
```

C:\Cora\Disciplinas\INF01202

```
While - soma = 2530
Do-while - soma = 2530
For - soma = 2530

Pressione qualquer tecla para continuar. . .
```

```
/* soma pares e mostra equivalências entre for, while e do while: */
#include <stdio.h>
#include <stdlib.h>
int main ( )
{
    system("color f1"); // muda cor neste comando!!!
    int par, soma;
    // usando while:
    soma = 0;
    par = 10; // valor inicial
    while (par <=100)
    {
        soma = soma + par;
        par=par + 2;
    }
    printf("\n\nWhile - soma = %d", soma);
    // usando do while:
    soma = 0;
    par = 10; // valor inicial
    do
    {
        soma = soma + par;
        par=par + 2;
    } while (par <=100);
    printf("\n\nDo-while - soma = %d", soma);
    // usando for:
    soma = 0;
    for ( par = 10; par <=100; par=par + 2)
        soma = soma + par;
    printf("\n\nFor - soma = %d\n\n", soma);
    system("pause");
    return 0;
}
```

2 - Escreva um programa que gera números menores que 5000, obtidos a partir do produto de 3 números pares consecutivos:

Objetivo:

Obter números < 5000 e resultante do produto de 3 números pares consecutivos.

Condições:

número < 5000

número = (n-2) * n * (n+2)

Logo:

(n-2) * n * (n+2) < 5000

```
/* Gera todos os números < 5000 resultante do produto de pares consecutivos */
#include <stdlib.h>
#include <stdio.h>
int main ()
{
    system("color f1"); // muda cor neste comando!!!
    int nro, par;
    // controle do for: cabeçalho - só variáveis incluídas aqui!!!
    for (par=4; (par-2) * par * (par + 2) < 5000 ; par = par + 2)
    {
        nro = (par-2) * par * (par + 2);
        printf("%d ", nro);
    }
    printf("\n");
    system("pause");
    return 0;
}
```

C:\Cora\Disciplinas\INF012...

48 192 480 960 1680 2688 4032

Pressione qualquer tecla para continuar. . .

3 - Escreva um programa que calcula o fatorial de vários números lidos. Encerrar a partir de opção do usuário.

```
/* calcula fatorial
#include <stdio.h>
int main ( )
{
    int num, fat, i, resposta;
    do // repete pelo menos 1 vez, até reposta = 1:
    {
        printf("\nDigite um número positivo: ");
        scanf("%d", &num);
        while (num < 0) // consiste número só vai adiante se válido:
        {
            printf("\nNúmero positivo ! - Digite outro: ");
            scanf("%d", &num);
        }
        fat = 1;
        for (i = num; i >= 2; i--)
            fat = fat * i;
        printf("\nFatorial = %d", fat);
        printf("\nOutro? (1 = sim / 0 = nao) ");
        scanf("%d", &resposta);
        while (resposta < 0 || resposta > 1) // consiste resposta:
        {
            printf("\nResposta inválida - outra (1 = sim / 0 = nao) ");
            scanf("%d", &resposta);
        }
    } while (resposta == 1);
    return 0;
}
```

4 - Foram entrevistados alunos de uma universidade.

De cada um deles foram colhidas as seguintes informações:

- o código do curso que frequenta (1-engenharia; 2-computação; 3-administração; 9-encerra pesquisa),
 - a idade. (se não houver encerrado a pesquisa)
- Faça um programa que processe estes dados e que informe:
- número de alunos por curso;
 - número de alunos com idade entre 20 e 25 anos, por curso;
 - qual a idade do aluno mais moço, qual a do mais velho e curso(s) correspondente(s).
 - qual o curso com menor média de idade.

// Estatísticas baseadas na idade dos alunos de 3 cursos:

```
#include <stdio.h>
#include <stdlib.h>
int main ( )
{
    // declarações das variáveis utilizadas no programa:
    int idade, curso; // 1-engenharia; 2-computacao; 3-administracao
    int al_1, al_2, al_3; // alunos por curso
    int al_2025_1, al_2025_2, al_2025_3; // por faixa etária
    int menos_id, menos_id_1, menos_id_2, menos_id_3; // menor idade geral e por curso
    int mais_id, mais_id_1, mais_id_2, mais_id_3; // maior idade geral e por curso
    float menor_media, med_id_1, med_id_2, med_id_3; // média das idades

    // inicializações:
    system("color f1"); // altera cores da tela de execução
    al_1=al_2=al_3=al_2025_1=al_2025_2=al_2025_3=0; // inicialização de contadores
    menos_id=menos_id_1=menos_id_2=menos_id_3=99; // inicialização de menor idade
    mais_id=mais_id_1=mais_id_2=mais_id_3=0; // inicialização de maior idade
    med_id_1=med_id_2=med_id_3=0; // para acumular as idades aqui mesmo
}
```

Algoritmo Estatística - usando laço controlado pelo while

{efetua contagem de alunos, totais e por curso}

```
1. Início
2. Efetua inicializações
3. Ler curso {leitura única e inicial, para saber se existem dados}
3. Enquanto (curso <> 9)
3.1     Início
3.2     Ler idade
3.4     Opção curso de
3.4.1         1 : Efetua contagens
3.4.2         2 : Efetua contagens
3.4.3         3 : Efetua contagens
3.4.4     Fim opção
3.5     Ler curso {obtem próximo curso ou informação de encerramento}
3.6     Fim enquanto
4. Informa totais por curso
5. Informa mais moço e curso(s)
6. Informa mais velho e curso(s)
7. Fim
```