

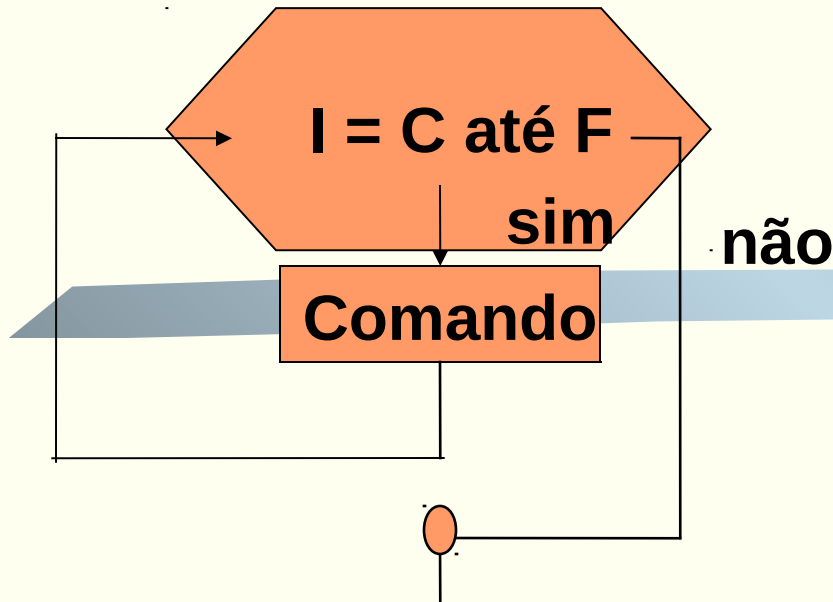
# ESTRUTURAS DE CONTROLE

## ESTRUTURAS DE REPETIÇÃO EM LINGUAGEM C



# Repetição Contada

Fluxograma



Linguagem C

```
For (i=C; i<=F; i++) {  
    comando;  
}
```

# Comando for

```
for (inicialização; teste; incremento)  
instrução;
```

```
for (inicialização; teste; incremento)  
{  
instrução_1;  
instrução_2;  
...  
instrução_n;  
}
```

# Comando for

## Exemplos:

```
for(i = 1; i<=10; i++)  
    printf("%d ", i);    ⇒ 1 2 3 4 5 6 7 8 9 10
```

```
for(i = 1; i<=10; i=i+1)  
    printf("%d ", i);    ⇒ 1 2 3 4 5 6 7 8 9 10
```

```
for(i = 10; i>=1; i--)  
    printf("%d ", i);    ⇒ 10 9 8 7 6 5 4 3 2 1
```

# Comando for

## Exemplos:

```
for(i = 1; i<=10; i+=2)
    printf("%d ", i);           ⇒ 1 3 5 7 9
```

```
for(i = 10; i>=1; i=i-3)
    printf("%d ", i);          ⇒ 10 7 4 1
```

```
for(i = -10; i<=10; i=i+5)
    printf("%d ", i);          ⇒ -10 -5 0 5 10
```

# Comando for

## Exemplos:

```
//Usando contador float  
float k;
```

```
for(k = 0; k<=1; k += 0.1)  
    printf("%3.1f ", k);
```

⇒ 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

```
//Usando contador char
```

```
char letra;
```

```
for(letra = 'A'; letra<='Z'; letra++)  
    printf("%c ", letra);
```

⇒ A B C ....X Y Z

```
for(letra = 'z'; letra>='a'; letra--)
```


```
    putchar(letra);
```

⇒ z y x ....c b a

# Comando for

// Lê um caractere e imprime o seguinte

```
int main()
{
    char ch;
    for(ch=getchar(); ch!='X'; ch=getchar())
        printf("%c", ch+1);
}
```



**Como sair do loop?**

# Comando for

- Qualquer uma das três partes do “for” pode ser omitida.
- Todavia, o ponto-e-vírgula deve estar presente.
- Se as expressões de inicialização e de incremento forem omitidas, elas apenas serão desconsideradas.
- Porém, se a expressão de teste for omitida, a mesma é considerada permanentemente verdadeira.
  - ✓ O que acontece nessa situação?



# Comando for

Exemplos: Considere `int i, sum=0`. Os trechos abaixo são equivalentes.

```
for(i=1; i<=10; i++)  
    sum += i;
```

```
i=1  
for(; i<= 10;i++ )  
    sum += i;
```

```
i=1;  
for(; i<= 10; )  
    sum += i++;
```

No entanto, há um loop infinito no código abaixo.

```
i=1; sum=0;  
for(; ; )  
    sum += i++;
```

# Comando for

- Qualquer expressão do comando “for” pode conter várias instruções separadas por vírgula.
- O operador vírgula usado no “for” permite múltiplas inicializações e múltiplos processamento de índices.
- Um par de expressões separado por vírgulas é avaliado da esquerda para a direita.

# Comando for

Exemplos: Considere `int i, sum=0`. Os trechos abaixo são equivalentes.

```
sum = 0;
for(i=1; i<=10; i++)
    sum += i;
```

```
for(sum=0, i=1; i<= 10; i++ )
    sum += i;
```

```
for(sum=0, i=1; i<= 10; sum+=i, ++i )
    ;
```

O trecho abaixo **NÃO** é equivalente  
`for(sum=0, i=1; i<= 10; ++i, sum+=i )`  
;

# Comando for

Exemplo:

```
//Usando mais de um contador
```

```
int main()
```

```
{
```

```
    int i,j;
```

```
    printf("\n\n(i,j)=");
```

```
    for(i = 1, j=1; i<=5, j<=5; i++, j++)
```

```
        printf("(%d,%d) ", i, j);
```

```
    system("PAUSE");
```

```
    return 0;
```

```
}
```

(i,j)=(1,1) (2,2) (3,3) (4,4) (5,5)

# Comando for

## Exemplo:

//imprime os números de 0 a 498 em incremento de 2

```
int main()
```

```
{
```

```
    int x,y;
```

```
    for(x=0, y=0; x+y < 500; x=x+1, y=y+1)
```

```
        printf("%d ",x+y);
```

```
        system("PAUSE");
```

```
    return 0;
```

```
}
```

0 2 4 6 8 10 12 ....494 496 498

# Repetição Contada

## Exemplo

**Desenvolver algoritmo para o problema:**

Dado um conjunto de  $N$  números, calcular a média aritmética dos mesmos

```
int main(){
  int i, n;
  float s=0, x;
  printf("n="); scanf("%d",&n);
  for(i=1; i <= n; i++){
    printf("X="); scanf("%f",&x);
    s = s + x;
  }
  printf("média=%.2f",s/n);
  return 0;
}
```

quantidade de notas

nota

# Repetição Contada

## Exemplo

**Desenvolver algoritmo para o problema:**

Ler um número inteiro  $\geq 0$  e calcular seu fatorial.



```
int main(){
  int i, n, fat;
  printf("n=") Inicializa o multiplicador
  fat = 1;
  for(i=1; i <= n; i++){
    fat = fat*i;
  }
  printf("%d!=%d",n, fat);
  return 0;
}
```

# Exercícios

1. Faça um programa que recebe um valor inteiro  $n$  e exibe os números pares a partir de  $n$ . Exemplo:
  - $N = 10$
  - **Saídas:** 0, 2, 4, 6, 8, 10.
2. Construa um código que apresente os valores inteiros maiores que 10 e menores que 25.
3. Faça um programa que recebe um valor inteiro  $n$  e exiba os números, a partir de  $n$ , de trás para frente em um intervalo de 3 em 3, Exemplo:
  - $N = 10$
  - **Saídas:** 10 7 4 1

# Exercícios

4. Faça um programa que leia um valor N inteiro e positivo, calcule e mostre o valor de E conforme a fórmula a seguir:

$$E = 1 + 1/1! + 1/2! + 1/3! + \dots + 1/N!$$

5. Faça um programa que receba números inteiros maiores que 1. A cada valor recebido, exiba um mensagem dizendo se o número é primo ou não.

# Exercícios

6. Faça um programa que receba valores inteiros positivos , calcule e mostre:

A soma dos números digitados.

A quantidade de números digitados.

A média dos números digitados.

O maior número digitado.

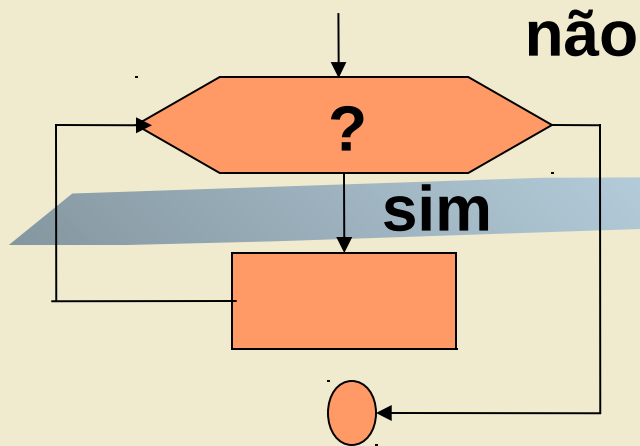
O menor número digitado.

A média dos números pares.

A porcentagem dos números ímpares entre todos os números digitados.

# Repetição com Teste no Início

Fluxograma



Linguagem C

```
while (condição) {  
    comando;  
}
```

# Comando while

```
while (expressão de teste)  
    instrução;
```

```
while (expressão de teste)  
{  
    instrução_1;  
    instrução_2;  
    ...  
    instrução_n;  
}
```

# Comando while

- “while” pode substituir o comando “for” da seguinte maneira:

Inicialização da variável de teste

```
while(teste)
```

```
{
```

```
    Incremento da variável de teste;
```

```
    ....
```

```
}
```

# Comando while

- Há equivalência entre comandos for e while

```
for (expr1; expr2; expr3)
    instrução1;
Instrução2;
```

```
expr1;
while(expr2){
    instrução1;
    expr3;
}
Instrução2;
```



# Comando while

## Exemplo 1

```
int main()
{
    int conta=0;
    int total=0;
    while(conta<10)
    {
        total+=conta;
        printf("conta=%d, total=%d\n", conta, total);
        conta++;
    }
    system("pause");
}
```

# Comando while

**Exemplo 2:** Há algum problema nos trechos de código abaixo ?

```
while (++i<LIMIT) do{ //trecho de código 1
    j=2*i+3;
    printf(“%d\n”,j);
}
```

```
printf(“Input an integer: “); //trecho de código 2
scanf(“%d”, &n);
while (--n)
    ..... /* faça alguma coisa */
```

```
while ((c=getchar()) == ‘ ‘) //trecho de código 3
    ;
```

# Repetição com Teste no Início

## Exemplo

```
int main(){
    char sexo;
    printf("sexo:"); scanf("%c",&sexo);
    while((sexo!='F')&&(sexo!='f')&&(sexo!='M')&&(sexo!='m')){
        printf("erro");
        printf("sexo:"); scanf("%c",&sexo);
    }
    printf("sexo:%c",sexo);
    return 0;
}
```

# Repetição com Teste no Início

Esta variável de controle deve ter um valor conhecido. Neste caso ela foi lida antes.

```
int main(){  
    char sexo;  
    printf("sexo:"); scanf("%c",&sexo);
```

laço

```
while((sexo!='F')&&(sexo!='f')&&(sexo!='M')&&(sexo!='m')){  
    printf("erro");  
    printf("sexo:"); scanf("%c",&sexo);  
}
```

```
printf("sexo:%c",sexo);  
return 0;  
}
```

A variável de controle deve ter seu valor modificado dentro do "laço". Neste caso a variável foi lida novamente.

Repetição com Teste no Início

# Contador/Acumulador **Exemplo**

**Desenvolver um programa para resolver o seguinte problema:**

- Ler um conjunto de valores correspondentes aos pontos que alunos obtiveram em um teste.
- Quando o valor fornecido for um número negativo, isto é um sinal que não existem mais pontos para serem lidos.
- Contar e escrever quantos alunos fizeram o teste.
- Contar e escrever quantos alunos tiveram nota baixa ( $PONTOS < 50$ )
- Contar e escrever quantos alunos tiveram nota alta ( $100 \leq PONTOS \leq 150$ )

Repetição com Teste no Início

# Contador/Acumulador **Exemplo**

**Desenvolver um programa para resolver o seguinte problema:**

- Ler um conjunto de notas correspondentes aos pontos que foram atribuídos a um teste.
- Quando o valor for zero ou negativo, isto indica que o aluno não fez o teste e não tem mais pontos para ser considerado.
- Contar e escrever o número de alunos que fizeram o teste.
- Contar e escrever quantos alunos tiveram nota baixa ( $PONTOS < 50$ ).
- Contar e escrever quantos alunos tiveram nota alta ( $100 \leq PONTOS \leq 150$ ).

São necessários  
3 contadores

```
int main(){
  int n,altas,baixas, pontos;
  n= 0;
  altas = 0;
  baixas = 0;
  printf("pontos:");scanf("%d",&pontos);
  while(pontos ≥ 0){
    if(pontos ≥ 100){
      altas = altas + 1;
    }
    else if (pontos < 50){
      baixas = baixas +1;
    }
    n=n+1;
    printf("pontos:");scanf("%d",&pontos);
  }
  printf("%d,%d,%d",n,baixas,altas);
  return 0;
}
```



3 contadores

# Repetição com Teste no Início

## Exemplo

**Desenvolver programa para resolver o problema:**

Ler N números reais (o valor de N também deve ser lido), sendo impressa a média desses números.



```
int main(){
    int i,j,n;
    float num, media;
    printf("n:");scanf("%d",&n);
    media = 0;
    i = 1;
    while(i ≤ n){
        printf("num:");
        scanf("%d",&num);
        media = media + num;
        i = i +1;
    }
    media = media /n;
    printf("media:%.2f",media);
    return 0;
}
```

```
int main(){
    int i,j,n;
    float num,media;
    printf("n:");
    media = 0;
    i = 1;
    while(i ≤ n){
        printf("num. ");
        scanf("%d",&num);
        media = media + num;
        i = i + 1;
    }
    media = media /n;
    printf("media:%.2f",media);
    return 0;
}
```

i é um contador  
que determina a  
parada

i é declarado  
como inteiro

i é  
inicializado

i é incrementado de 1

```
int main(){
    int i,j,n;
    float num, media;
    printf("n:");scanf("%d",&n);
    media = 0;
    i = 0;
    while(i ≤ n-1){
        i = i + 1;
        printf("num:");
        scanf("%d",&num);
        media = media + num;
    }
    media = media /n;
    printf("media:%.2f",media);
    return 0;
}
```

i é um contador que determina a parada

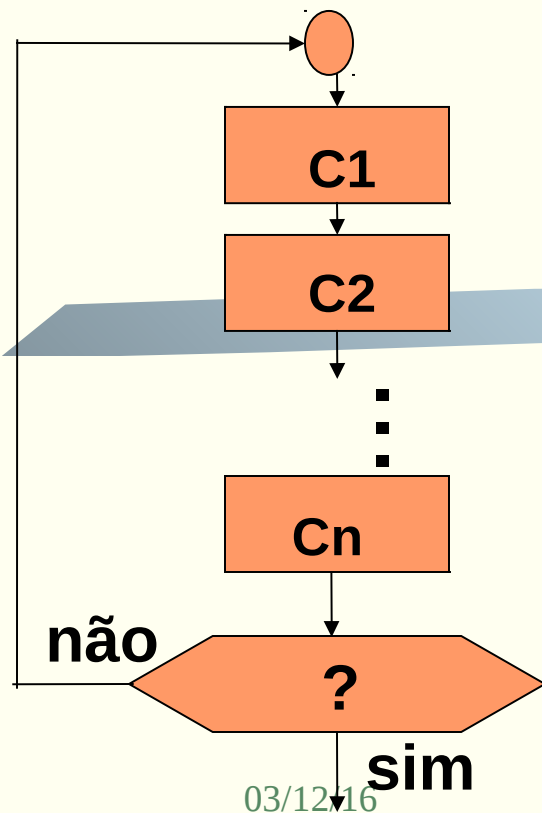
i é inicializado

i é incrementado de 1

Outra maneira de contar os números

# Repetição com Teste no Final

Fluxograma

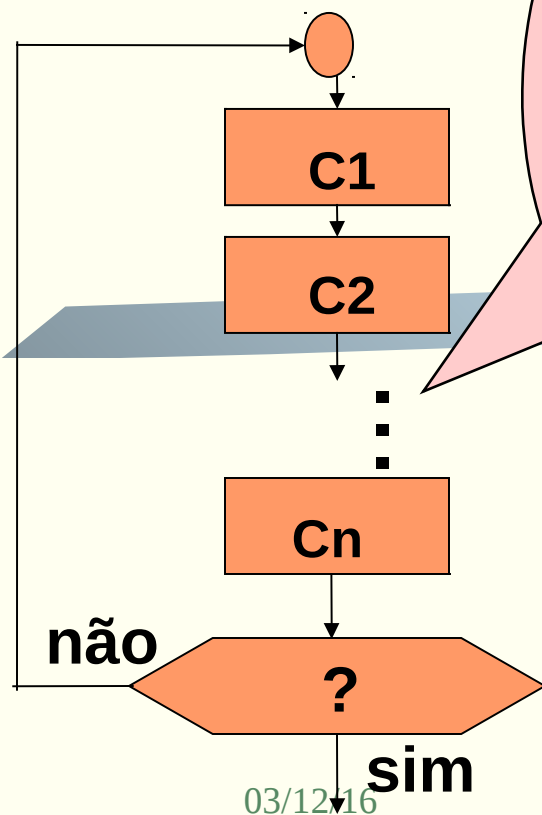


Linguagem C

```
do{  
    comando 1;  
    comando 2;  
  
    comando n;  
while(<condição>);
```

# Repetição com Teste no Final

Fluxograma



Os comandos são executados pelo menos uma vez

Linguagem C

```
do{  
  comando 1;  
  comando 2;  
  
  comando n;  
while(<condição>);
```

# Comando do-while

```
do  
{  
    instrução;  
} while(expressão de teste);
```

- Permite executar o bloco “do-while” mesmo se o teste for falso no início.
- Normalmente pouco utilizado.

# Comando do-while

## Exemplo 1:

```
//Soma os valores inteiros até receber valor 0
```

```
int i=0;
```

```
int sum=0;
```

```
do{
```

```
    sum += i;
```

```
    scanf("%d", &i);
```

```
}while (i>0);
```

```
printf("Somatorio eh: %d", sum);
```

# Comando do-while

**Exemplo 2:** Recebe apenas inteiros positivos

```
int n, error;
do{
    printf("Entre com valor inteiro positivo: ");
    scanf("%d", &n);
    if(error = (n<=0))
        printf("\nERROR: Digite novamente!\n\n");
}while (error);
```



# Comando do-while

**Exemplo 3:** Código realizado com o comando WHILE.

```
char sair;
//Pedindo pra sair em um while
printf("\nPede pra sair!!! (digite S):");
sair = toupper(getchar());
while(sair!='S'){
    printf("\nPede pra sair!!! (digite S):");
    sair = toupper(getchar());
}
printf("\nSaiu!!!");
```

# Comando do-while

**Exemplo 4:** Código realizado com o comando Do-While

```
char sair;  
//Pedindo pra sair com do-while  
do{  
    printf("\nPede pra sair!!! (digite S):");  
    sair = toupper(getchar());  
}while(sair!='S');  
printf("\nSaiu!!!");
```

# Repetição com Teste no Final Exemplo

**Desenvolver algoritmo para o problema:**

Calcular e exibir o valor da série.

$$S = \frac{1}{1} + \frac{3}{2} + \frac{5}{3} + \frac{7}{4} + \dots + \frac{99}{50}$$

```
int main(){
```

```
int n,d;
```

```
float s;
```

```
s= 0;
```

```
n = 1;
```

```
d = 1;
```

```
do{
```

```
s = s +(float)  $\frac{n}{d}$  ;
```

```
n = n+2;
```

```
d = d+1;
```

```
}while (d <= 50);
```

```
printf("s=%.2f",s);
```

```
return 0;
```

```
}
```

Inicializa a somatória

Inicializa o numerador

Inicializa o denominador

Calcula os termos da série

Calcula o próximo numerador

Calcula o próximo denominador

# Comando break

- Utilizado no corpo de qualquer estrutura de laço.
- Causa a saída imediata do laço, desviando o programa para a próxima instrução após o laço atual.
- Se estiver em laços aninhados, o **break** afetará somente o laço que o contém e seus laços internos.
- No caso de laços como **for**, **while** e **do-while**, o laço é interrompido e os comandos do programa são retomados a partir da primeira linha fora do laço.
- No caso do **switch**, a execução da sequência de comandos é interrompida.

# Comando break

## Exemplo:

```
int x, sum;
sum=0;
while(1){
scanf("%f",&x);
if(x<0.0)
break;
sum += sqrt(x);
}
printf("sum= %f\n", sum);
```

# Comando continue

- Força a execução da próxima iteração do laço, não executando o código que vem a seguir.
- Esse comando ocorre apenas nos comandos “for”, “while” e “do-while”.
- Esse comando deve ser evitado, pois dificulta a leitura e manutenção do código.

# Comando continue

No caso do “while” e “do-while” a execução é desviada para o teste condicional e depois segue para o corpo do laço.

Interrompe a sequência de execuções dentro do laço,  
Verifica a condição.

No caso do “for”, o desvio é feito para o incremento, seguido pelo teste condicional e corpo do laço.

Interrompe a sequência de execuções dentro do laço,  
Incrementa ou decrementa o contador,  
Verifica a condição.



# Comando continue

## Exemplo:

```
int main(){
    int valor;
    for(valor=1; valor<=10; valor++){
        if(valor%2)
            continue;
        printf("%d ", valor);
    }
    return 0;
}
```

# Exercícios

Faça um programa que lê 10 números inteiros, sendo que desses números deve-se mostrar apenas os números menores que 5 e maiores que 30. Caso não tenha, deve apresentar uma mensagem informando que não há números no intervalo definido.

Faça um programa que receba números inteiros maiores que 1. A cada valor recebido, exiba uma mensagem dizendo se o número é primo ou não.

# Exercícios

Faça um programa que receba valores inteiros positivos, calcule e mostre:

- A soma dos números digitados.
- A quantidade de números digitados.
- A média dos números digitados.
- O maior número digitado.
- O menor número digitado.
- A média dos números pares.
- A percentagem dos números ímpares entre todos os números digitados.