

ESTRUTURA DE REPETIÇÃO

ESTRUTURA DE REPETIÇÃO PARA NÚMERO INDEFINIDO DE REPETIÇÕES E TESTE NO INÍCIO (ESTRUTURA ENQUANTO - FAÇA)

Essa estrutura de repetição é utilizada quando *não* se sabe o número de vezes em que um trecho do algoritmo deve ser repetido, embora também possa ser utilizada quando se sabe esse número.

Existem situações em que o teste condicional da estrutura de repetição, que fica no início, resulta em um valor falso logo na primeira comparação. Nesses casos, os comandos de dentro da estrutura de repetição não serão executados.

ENQUANTO condição FAÇA comando 1

Enquanto a condição for verdadeira, o comando1 será executado.

```
ENQUANTO condição FAÇA INÍCIO
    comando 1;
    comando 2 ;
    comando 3
FIM
```

Enquanto a condição for verdadeira, os comando1, comando2, comando3 serão executados.

Exemplo:

```
X ← 1
Y ← 5
ENQUANTO X < Y FAÇA
    INÍCIO
        X ← X+2;
        Y ← Y+1;
    FIM
```

| X | Y | |
|---|---|------------------|
| 1 | 5 | Valores Iniciais |

| | | |
|---|---|--|
| 3 | 6 | |
| 5 | 7 | Valores obtidos dentro da Estrutura de Repetição |
| 7 | 8 | |
| 9 | 9 | |

Portanto, no trecho do algoritmo anterior, os comandos que se localizam dentro da estrutura de repetição serão repetidos quatro vezes.

ESTRUTURA DE REPETIÇÃO PARA NÚMERO INDEFINIDO DE REPETIÇÕES E TESTE NO FINAL (ESTRUTURA REPITA - ATÉ)

Essa estrutura de repetição é utilizada quando *não* se sabe o número de vezes em que um trecho do algoritmo deve ser repetido, embora também possa ser utilizada quando sabe esse número.

A diferença entre a estrutura **ENQUANTO** e a estrutura **REPITA** é que na estrutura REPITA os comandos serão repetidos pelo menos uma vez, já que a condição se encontra no final.

REPITA

Comandos ;

ATÉ condição

Repita os **comandos** até a **condição** se tornar verdadeira.

Exemplo:

```

X ← 1
Y ← 5
REPITA
  X ← X + 2;
  Y ← Y + 1;
ATÉ X >= Y

```

| | | |
|---|---|------------------|
| X | Y | |
| 1 | 5 | Valores Iniciais |

| | | |
|---|---|--|
| 3 | 6 | |
| 5 | 7 | Valores obtidos dentro da Estrutura de Repetição |
| 7 | 8 | |
| 9 | 9 | |

Portanto, no trecho do algoritmo anterior, os comandos que se localizam dentro da estrutura de repetição serão repetidos quatro vezes.

ESTRUTURA DE REPETIÇÃO PARA NÚMERO DEFINIDO DE REPETIÇÕES (ESTRUTURA PARA - FAÇA)

Essa estrutura de repetição é utilizada quando se sabe e o número de vezes em que um trecho do algoritmo deve ser repetido.

PARA $I \leftarrow$ *παλορ ινιχιαλ* ATE *παλορ φιναλ* FAÇA
 χομανδο 1;

O **comando1** será executado utilizando a variável **I** como controle, cujo conteúdo vai variar do **valor inicial** até o **valor final**, de 1 em 1, incrementando automaticamente.

```

PARA J ← valor INICIAL valor FINAL FAÇA
  INÍCIO
    Comando1
    Comando2
  FIM

```

Os **comando1** e **comando2** serão executados utilizando a variável **J** como controle, cujo conteúdo vai variar do **valor inicial** até o **valor final**, de 1 em 1, incrementando automaticamente.

Exemplo:

```
PARA I ← 1 ATÉ 10 FAÇA  
  Comando1;
```

O comando1 será executado dez vezes.

ESTRUTURA DE REPETIÇÃO EM PASCAL

ESTRUTURA DE REPETIÇÃO FOR

Essa estrutura de repetição é utilizada quando se sabe o número de vezes em que um trecho do programa deve ser repetido.

```
FOR I:= valor inicial TO valor final DO  
  comando;
```

o comando será executado utilizando a variável **I** como controle, cujo conteúdo vai variar do valor inicial até o valor final, de 1 em 1, incrementando automaticamente.

```
FOR J:= valor inicial TO valor final  
DO  
BEGIN  
  Comando1;  
  Comando2;  
END;
```

Os comando1 e comando2 serão executados utilizando a variável **J** como controle, cujo conteúdo vai variar do valor inicial até o valor final, de 1 em 1, incrementando automaticamente.

```
FOR K := valor inicial DOWNTO valor final DO  
comando;
```

O comando será executado utilizando a variável **K** como controle, cujo conteúdo vai variar do valor inicial até o valor final, de 1 em 1, decrementando automaticamente.

```
FOR H:=.valor inicial DOWNTO valor final DO
```

```
BEGIN
```

```
Comando1;
```

```
comando2;
```

```
comando3;
```

```
END;
```

OS comando1, comando2 e comando3 serão executados utilizando a variável H como controle, cujo conteúdo vai variar do valor inicial até o valor final, de 1 em 1, decrementando automaticamente.

ESTRUTURA DE REPETIÇÃO WHILE

WHILE condição DO

comando;

Enquanto a condição for **verdadeira**, o comando será executado.

WHILE condição DO

BEGIN

Comando1;

Comando2;

END;

Enquanto a condição for verdadeira, os comando1 e comando2 serão executados.

ESTRUTURA DE REPETIÇÃO “REPEAT”

REPEAT

comandos;

UNTIL condição;

EXERCÍCIOS

1. Um funcionario de uma empresa recebe aumento salarial anualmente.

Sabe-se que:

a) esse funcionario foi contratado em 1995, com salário inicial de R\$ 1.000,00;

b) em 1996 recebeu aumento de 1,5% sobre seu salário inicial;

c) a partir de 1997 (inclusive), os aumentos salariais sempre corresponderam ao dobro do percentual do ano anterior.

Faça um programa que determine o salário atual desse funcionario.

1ª SOLUÇÃO UTILIZANDO ESTRUTURA FOR:

```
PROGRAM EX1;
USES CRT;
VAR      i, ano_atual: INTEGER;
salario, novo_salario, percentual: REAL;
BEGIN
CLRSCR;
WRITELN('Digite o ano atual');
READLN(ano_atual);
salario := 1000;
percentual := 1.5/100;
novo_salario := salario + percentual * salario;
FOR i := 1997 TO ano_atual DO
BEGIN
percentual := 2 * percentual;
novo_salario := novo_salario + percentual * novo_salario;
END;
WRITELN('Novo salario = ',novo_salario:5:2);
READKEY;
END.
```

2ª SOLUÇÃO UTILIZANDO ESTRUTURA WHILE:

```
PROGRAM EX1;
```

```

USES CRT;
VAR      i, ano_atual: INTEGER;
salario, novo_salario, percentual: REAL;
BEGIN
CLRSCR;
WRITELN('Digite o ano atual');
READLN(ano_atual);
salario := 1000;
percentual := 1.5/100;
novo_salario := salario + percentual * salario;
i := 1997;
WHILE i <= ano_atual DO
BEGIN
percentual := 2 * percentual;
novo_salario := novo_salario + percentual * novo_salario;
i := i + 1;
END;
WRITELN('Novo sal rio = ',novo_salario:5:2);
READKEY;
END.

```

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

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

1ª SOLUÇÃO UTILIZANDO ESTRUTURA FOR:

```

PROGRAM EX2;
USES CRT;
VAR n, i, j: INTEGER;
    e, fat: real;
BEGIN
CLRSCR;
WRITELN('Digite o valor de N');
READLN(n);
e := 1;

```

```

FOR i := 1 TO n DO
  BEGIN
    fat := 1;
    FOR j := 1 TO i DO
      BEGIN
        fat := fat * j;
      END;
    e := e + 1/fat;
  END;
WRITELN('Valor de E = ',e:5:2);
READKEY;
END.

```

2ª SOLUÇÃO UTILIZANDO ESTRUTURA REPEAT:

```

PROGRAM EX2;
USES CRT;
VAR n, i, j: INTEGER;
    e, fat: REAL;
BEGIN
  CLRSCR;
  WRITELN('Digite o valor de N');
  READLN(n);
  e := 1;
  i := 1;
  REPEAT
    j := 1;
    fat := 1;
    REPEAT
      fat := fat * j;
      j := j + 1;
    UNTIL j > i;
    i := i + 1;
    e := e + 1/fat;
  UNTIL i > n;
  WRITELN('Valor de E = ',e:5:2);
  READKEY;
END.

```

3. Faça um programa que leia um número N e que indique quantos valores inteiros e positivos devem ser lidos a seguir. Para cada número lido, mostre

uma tabela contendo o valor lido e o fatorial desse valor.

1ª SOLUÇÃO UTILIZANDO ESTRUTURA FOR:

```
PROGRAM EX3;
USES CRT;
VAR n, num, i, j:INTEGER;
    fat: REAL;
BEGIN
CLRSCR;
WRITELN('Digite a quantidade de números que serão lidos');
READLN(n);
FOR i := 1 TO n DO
BEGIN
WRITELN;
WRITELN('Digite o ', i, ' número');
READLN(num);
fat := 1;
FOR j := 1 TO num DO
BEGIN
fat := fat * j;
END;
WRITELN('Fatorial de ', num, ' = ', fat:5:2);
END;
READKEY;
END.
```

2ª SOLUÇÃO UTILIZANDO ESTRUTURA WHILE:

```
PROGRAM EX3;
USES CRT;
VAR n, num, i, j:INTEGER;
    fat: REAL;
BEGIN
CLRSCR;
WRITELN('Digite a quantidade de números que serão lidos');
READLN(n);
i := 1;
WHILE i <= n DO
```

```

BEGIN
WRITELN;
WRITELN('Digite o ', i, ' numero');
READLN(num);
fat := 1;
j := 1;
WHILE j <= num DO
    BEGIN
        fat := fat * j;
        j := j + 1;
    END;
WRITELN('Fatorial de ', num, ' = ', fat:5:2);
i := i + 1;
END;
READKEY;
END.

```

4. Faça um programa que leia cinco pares de valores (a,b), todos inteiros e positivos, um de cada vez. Mostre os números inteiros pares de a até b (inclusive).

1ª SOLUÇÃO UTILIZANDO ESTRUTURA FOR:

```

PROGRAM EX4;
USES CRT;
VAR cont, a, b, i: INTEGER;
BEGIN
CLRSCR;
FOR cont := 1 TO 5 DO
    BEGIN
        WRITE('Digite o primeiro numero do intervalo: ');
        READLN(a);
        WRITE('Digite o segundo numero do intervalo: ');
        READLN(b);
        IF a < b
        THEN BEGIN
            WRITE('Os pares do ', cont, ' intervalo sao: ');
            FOR i := a TO b DO
                BEGIN

```

```
        IF (i MOD 2) = 0
        THEN WRITE(i, ' ');
    END;
    WRITELN;
    WRITELN;
    END
ELSE IF a > b
    THEN BEGIN
        WRITE('Os pares do ',cont,' intervalo sao: ');
        FOR i := a DOWNTO b DO
            BEGIN
                IF (i MOD 2) = 0
                THEN WRITE(i, ' ');
            END;
        WRITELN;
        WRITELN;
        END
    ELSE IF (a = b) AND (a MOD 2 = 0)
        THEN BEGIN
            WRITE('O numero par do ',cont,' intervalo ,: ');
            WRITELN(a);
            WRITELN;
            WRITELN;
            END
        ELSE BEGIN
            WRITELN('Neste intervalo nao existe numero par');
            WRITELN;
            WRITELN;
            END;
    END;
    READKEY;
    END.
```

2ª SOLUÇÃO UTILIZANDO ESTRUTURA REPEAT:

```
PROGRAM EX4;
USES CRT;
VAR cont, a, b, i: INTEGER;
BEGIN
CLRSCR;
cont := 0;
REPEAT
WRITE('Digite o primeiro numero do intervalo: ');
READLN(a);
WRITE('Digite o segundo numero do intervalo: ');
READLN(b);
cont := cont + 1;
IF a < b
THEN BEGIN
WRITE('Os pares do ',cont,' intervalo sao: ');
REPEAT
IF (a MOD 2) = 0
THEN WRITE(a,' ');
a := a + 1;
UNTIL a > b;
WRITELN;
WRITELN;
END
ELSE IF a > b
THEN BEGIN
WRITE('Os pares do ',cont,' intervalo sao: ');
REPEAT
IF (b MOD 2) = 0
THEN WRITE(b,' ');
b := b + 1;
UNTIL b > a;
WRITELN;
WRITELN;
END
ELSE IF (a = b) AND (a MOD 2 = 0)
THEN BEGIN
WRITE('O numero par do ',cont,' intervalo ,: ');
WRITELN(a);
WRITELN;
WRITELN;
END
ELSE BEGIN
WRITELN('Neste intervalo nao existe numero par');
```

```

        WRITELN;
        WRITELN;
        END;
UNTIL CONT = 5;
READKEY;
END.

```

5. Faça um programa que leia dez conjuntos de dois valores, o primeiro representando o número do aluno e o segundo representando a sua altura em centímetros. Encontre o aluno mais alto e o mais baixo. Mostre o número do aluno mais alto e o número do mais baixo junto com suas alturas.

1ª SOLUÇÃO UTILIZANDO ESTRUTURA FOR:

```

PROGRAM EX5;
USES CRT;
VAR cont, num, num_maior, num_menor: INTEGER;
    alt, maior, menor: REAL;
BEGIN
CLRSCR;
FOR cont := 1 TO 10 DO
    BEGIN
    WRITE('Digite o numero do ', cont, ' aluno ');
    READLN(num);
    WRITE('Digite a altura do ', cont, ' aluno ');
    READLN(alt);
    WRITELN;
    IF cont = 1
    THEN BEGIN
        maior := alt;
        num_maior := num;
        menor := alt;
        num_menor := num;
    END
    ELSE BEGIN
        IF alt > maior
        THEN BEGIN
            maior := alt;
            num_maior := num;
        END;
    END;
    END;

```

```

        IF alt < menor
        THEN BEGIN
            menor := alt;
            num_menor := num;
        END;
    END;
END;
WRITELN('Maior altura = ',maior:4:2, ' e , do aluno
de numero ',num_maior);
WRITELN('Menor altura = ',menor:4:2, ' e , do aluno
de numero ',num_menor);
READKEY;
END.

```

2ª SOLUÇÃO UTILIZANDO ESTRUTURA WHILE:

```

PROGRAM EX5;
USES CRT;
VAR cont, num, num_maior,num_menor: INTEGER;
    alt, menor, maior: REAL;
BEGIN
    CLRSCR;
    cont := 1;
    WHILE cont <= 10 do
    BEGIN
        WRITE('Digite o numero do ',cont,'aluno ');
        READLN(num);
        WRITE('Digite a altura do ',cont,' aluno ');
        READLN(alt);
        WRITELN;
        IF cont = 1
        THEN BEGIN
            maior := alt;
            num_maior := num;
            menor := alt;
            num_menor := num;
        END
    ELSE BEGIN
        IF alt > maior
        THEN BEGIN
            maior := alt;

```

```

        num_maior := num;
        END;
    IF alt < menor
    THEN BEGIN
        menor := alt;
        num_menor := num;
        END;
    END;
cont := cont + 1;
END;
WRITELN('Maior altura = ', maior:4:2, ' e ', do_aluno
de_numero ', num_maior);
WRITELN('Menor altura = ', menor:4:2, ' e ', do_aluno
de_numero ', num_menor);
READKEY;
END.

```

8. Uma empresa possui dez funcionários com as seguintes características: código, número de horas trabalhadas no mês, turno de trabalho (M - Matutino, V - Vespertino ou N - Noturno), categoria (O - Operário ou G - Gerente), valor da hora trabalhada. Sabendo-se que essa empresa deseja informatizar sua folha de pagamento, faça um programa que:
- leia as informações dos funcionários, exceto o valor da hora trabalhada, não permitindo que sejam informados turnos nem categorias inexistentes. Trabalhar sempre com a digitação de letras maiúsculas;
 - calcule o valor da hora trabalhada, conforme a tabela a seguir.

| CATEGORIA | TURNO | VALOR DA HORA TRABALHADA |
|-----------|--------|--------------------------|
| G | N | 18% do salário mínimo |
| G | M ou V | 15% do salário mínimo |
| O | N | 13% do salário mínimo |
| O | M ou V | 10% do salário mínimo |

Adote o valor de R\$ 150,00 para o salário mínimo

- calcule o salário inicial dos funcionários com base no valor da hora trabalhada e no número de horas trabalhadas;
- calcule o valor do auxílio-alimentação recebido por funcionário de acordo com o seu salário inicial, conforme a tabela a seguir.

| SALÁRIO INICIAL | AUXÍLIO ALIMENTAÇÃO |
|-------------------------------|------------------------|
| Até R\$ 300,00 | 20% do salário inicial |
| Entre R\$ 300,00 e R\$ 600,00 | 15% do salário inicial |
| Acima de R\$ 600,00 | 5% do salário inicial |

e) mostre o código, número de horas trabalhadas, valor da hora trabalhada, salário inicial, auxílio-alimentação e o salário final (salário inicial + auxílio-alimentação).

1ª SOLUÇÃO UTILIZANDO ESTRUTURA FOR:

```

PROGRAM EX8;
USES CRT;
VAR cont, codigo: INTEGER;
    nht, valor, sal_min, sal_inicial, aux,
sal_final: REAL;
    turno, categoria: CHAR;
BEGIN
CLRSCR;
sal_min := 150;
FOR cont := 1 TO 10 DO
BEGIN
WRITELN('Digite o código do ', cont, 'funcionario');
READLN(codigo);
WRITELN('Digite o numero de horas trabalhadas do
', cont, 'funcionario');
READLN(nht);
WRITELN('Digite o turno de trabalho do
', cont, 'funcionario');
READLN(turno);
    WHILE (turno <> 'M') AND (turno <> 'V') AND
(turno <> 'N') DO
        BEGIN
            WRITELN('Turno invalido, digite novamente');
            READLN(turno);
        END;
WRITELN('Digite a categoria do
', cont, 'funcionario');
READLN(categoria);
    WHILE (categoria <> 'G') AND (categoria <> 'O')
DO

```

```

        BEGIN
            WRITELN('Categoria invalida, digite
novamente');
            READLN(categoria);
            END;
        IF categoria = 'G'
        THEN BEGIN
            IF turno = 'N'
            THEN valor := 18/100 * sal_min
            ELSE valor := 15/100 * sal_min;
            END
        ELSE BEGIN
            IF turno = 'N'
            THEN valor := 13/100 * sal_min
            ELSE valor := 10/100 * sal_min;
            END;
        sal_inicial := nht * valor;
        IF sal_inicial <= 300
        THEN aux := 20/100 * sal_inicial
        ELSE IF sal_inicial < 600
        THEN aux := 15/100 * sal_inicial
        ELSE aux := 5/100 * sal_inicial;
        sal_final := sal_inicial + aux;
        WRITELN('Codigo = ',codigo);
        WRITELN('Numero de horas trabalhadas =
',nht:5:2);
        WRITELN('Valor do hora trabalhada = ',valor:5:2);
        WRITELN('Salario inicial = ',sal_inicial:5:2);
        WRITELN('Auxilio comida = ',aux:5:2);
        WRITELN('Salario Final = ',sal_final:5:2);
    END;
    READKEY;
    END.

```

2ª SOLUÇÃO UTILIZANDO ESTRUTURA REPEAT:

```

PROGRAM EX8;
USES CRT;
VAR cont, codigo: INTEGER;
    nht, valor, sal_min, sal_inicial, aux,
sal_final: REAL;

```

```

        turno, categoria: CHAR;
BEGIN
CLRSCR;
sal_min := 150;
cont :=1;
REPEAT
WRITELN('Digite o codigo do ',cont,'funcionario');
READLN(codigo);
WRITELN('Digite o numero de horas trabalhadas do
',cont,'funcionario');
READLN(nht);
WRITELN('Digite o turno de trabalho do
',cont,'funcionario');
READLN(turno);
WHILE (turno <> 'M') AND (turno <> 'V') AND (turno
<> 'N') DO
    BEGIN
        WRITELN('Turno invalido, digite novamente');
        READLN(turno);
        END;
WRITELN('Digite a categoria do
',cont,'funcionario');
READLN(categoria);
WHILE (categoria <> 'G') AND (categoria <> 'O') DO
    BEGIN
        WRITELN('Categoria invalida, digite novamente');
        READLN(categoria);
        END;
IF categoria = 'G'
    THEN BEGIN
        IF turno = 'N'
            THEN valor := 18/100 * sal_min
            ELSE valor := 15/100 * sal_min;
            END
        ELSE BEGIN
            IF turno = 'N'
                THEN valor := 13/100 * sal_min
                ELSE valor := 10/100 * sal_min;
                END;
sal_inicial := nht * valor;
IF sal_inicial <= 300

```

```
    THEN aux := 20/100 * sal_inicial
    ELSE IF sal_inicial < 600
        THEN aux := 15/100 * sal_inicial
        ELSE aux := 5/100 * sal_inicial;
    sal_final := sal_inicial + aux;
    WRITELN('Codigo = ',codigo);
    WRITELN('Numero de horas trabalhadas =
',nht:5:2);
    WRITELN('Valor do hora trabalhada = ',valor:5:2);
    WRITELN('Salario inicial = ',sal_inicial:5:2);
    WRITELN('Auxilio comida = ',aux:5:2);
    WRITELN('Salario Final = ',sal_final:5:2);
cont := cont +1;
UNTIL CONT = 11;
READKEY;
END.
```