

Estrutura de programas Pascal

Program nome_programa; {opcional}

Uses Crt, printer, graph;

Const
 x=100;

Type {Para Definição de Tipos - pelo usuário. Poderá ser estudo adiante.}
 nome = string[80];

var
 a,b :integer;
 c: real;
 n: nome;

Begin
 {comandos}
End.

{
Tipos de dados no Pascal

Palavra reservada para o tipo: [faixa de valores do tipo]. Utilização de memória do tipo.

integer : [-32768 ... 32767]. Utiliza 2 bytes em memória.
real : [-1.0E-38 ... 1.0E38]. Utiliza 6 bytes em memória.
byte : [0 ... 255]. Utiliza 1 byte em memória.
char : Utiliza 1 byte em memória.
boolean: Utiliza 1 byte em memória.
String[n]: Utiliza (n+1) bytes em memória.
word: [0 ... 65535]. Utiliza 2 bytes em memória.
shortint: [-128...128]. Utiliza 1 byte em memória.
longint:[-2147483648 ... 2147483647]. Utiliza 4 bytes em memória.
Array[faixa] of tipo: utilizado para criação de vetores e matrizes. Será estudo adiante.

Correspondência entre os tipos de dados entre português estruturado e pascal

Numérico ⇔ integer, real, byte, word, shortint, longint

Literal ⇔ String[n], char (para um caractere)

Lógico ⇔ boolean

}

{

Operadores

+, -, *, /: operadores para tipos de dados reais.

+, -, *, /, DIV, MOD: operadores para tipos de dados inteiros.

DIV: Quociente
MOD: Resto

Forma de utilização: da mesma forma que os outros operadores - notação infixa.

X MOD Y
X DIV Y

}

{

Precedência de operadores

- 1 Parênteses
- 2 Operador unário (-): Ex. -(x)
- 3 Funções
- 4 Operador NOT
- 5 *, /, DIV, MOD, AND
- 6 +, -, OR, XOR
- 7 =, <>, >, <, >=, <=

}

{

Funções Matemáticas

Exp(x): constante euler (e) elevado a x
Abs(x): valor absoluto de x (módulo)
Arctan(x): arco tangente de x (x em radianos)
cos(x): co-seno de x (x em radianos)
sen(x): seno de x (x em radianos)
Frac(x): retorna a parte fracionária de x
int(x): retorna a parte inteira de x
Sqr(x): x^2
Sqrt(x): raiz quadrada de x (\sqrt{x})
Ln(x): logaritmo neperiano (ou natural) de x (base e - euler)
Chr(x): transforma ordinal em caractere
Ord(x): transforma caractere em ordinal
Round(x): converte real em inteiro arredondando
Trunc(x): converte real em inteiro sem arredondar
Pi: valor de π (π)= 3.1415926535897932385
Random(v): gera um número aleatório de zero até (v-1)

Como não existe uma função especial de potência no pascal, utiliza-se a seguinte expressão para elevar um número x ao número y .

$x^y ==> \exp(\ln(x)*y)$

pois:

$$x^y = e^{\ln(x) \cdot y}$$

$$\ln(x^y) = \ln(e^{\ln(x) \cdot y})$$

$$y \cdot \ln x = (\ln(x) \cdot y) \cdot \ln e; \text{ como } \ln e = 1$$

$$y \cdot \ln x = \ln(x) \cdot y$$

Logaritmos de outras bases

$$\log_b x = \frac{\ln x}{\ln b}$$

}
{

Comando de Leitura (entrada)

Leia a, b

No pascal:

Read(a, b); | Readln(a, b);

Read - lê o valor e não troca de linha

Readln - lê o valor e troca de linha

Readkey - lê um caractere (não necessita utilizar o "enter")

Ex.:

write('Digite uma tecla');

tecla:=Readkey;;

Comando de Escrita (saída)

Escreva "Digite um número"

No pascal:

write('Digite um número'); | writeln('Digite um número');

write - escreve a argumento e não troca de linha

writeln - escreve a argumento e troca de linha

writeln; deixa uma linha em branco

clrscr; {comando que limpa a tela do usuário (do computador). É necessário utilizar a *unit Crt*}

Comando de atribuição ($\leftarrow \Leftrightarrow :=$):

$a \leftarrow (x + 3)/y$

No pascal:

a:=(x+3)/y;

Comando Condicional (se...então...senão ⇔ if...then...else):

Forma geral:

se <condição> **então** <comandos>

senão <comandos>

if <condição> **then** <comandos>

else <comandos>

Exemplo:

Português Estruturado

se (a>b) E (b <10) então

início

 escreva a+b

 escreva a-b

fim

senão

início

 escreva a × b

 escreva a/b

fim

Pascal

if (a>b) and (b<10) then

begin

 writeln(a+b);

 writeln(a-b);

end {aquí não precisa ; (ponto e vírgula) pois aparece imediatamente antes do **else**}

else

begin

 writeln(a * b);

 writeln(a/b);

end;

Todo comando na linguagem Pascal deve ser encerrado por ; (ponto e vírgula), exceto quando o comando aparece imediatamente antes de **else** (if ... then ...else...). Existem algumas situações onde o ; (ponto e vírgula) é opcional.

Para leitura de múltiplos literais, isto é, ler vários valores literais para serem armazenados em várias variáveis do tipo *string*, é necessária a utilização de múltiplas chamadas de *readln*.

Read(nome); {funciona}

Readln(nome); {funciona}

Read(nome,ender); {não funciona}

Readln(nome, ender); {não funciona}

Readln(nome);

```

Readln(ender);
{funciona}
}

```

Estrutura de Repetição

Estrutura de repetição com número definido de repetições (*para, for*)
 Forma Geral:

Português Estruturado (*para até faça*)

Para <variável> \leftarrow <valorinicial> até <valorfinal> faça
 <comandos> {bloco de comandos}

Exemplo:

Para i \leftarrow 1 até 5 faça
início
 escreva i
 escreva i*i
fim

Pascal (estrutura *for ... to ... do*)

```

for i:= 1 to 5 do
begin
    write(i, ' : ');
    writeln(i*i);
end;

```

O resultado da execução desta parte de programa resultaria na impressão do seguinte texto como saída no monitor.

```

1 : 1
2 : 4
3 : 9
4 : 16
5 : 25

```

Exemplo:

Somatório dos primeiros n números inteiros positivos.

```

program somatorio;
uses crt;

var n,i, soma: integer;

begin
    clrscr;
    write('Digite um número: ');
    readln(n);
    soma:=0;
    for i:=1 to n do

```

```

begin
    soma:=soma+i;
end;
writeln('O somatorio dos primeiros ', n , ' numeros inteiros positivos e ', soma);
readkey;
end.

```

{ou de outra forma, sem utilizar repetição}

```

program somatorio;
uses crt;

var n, soma:real;
begin
    clrscr;
    write('Digite um numero: ');
    readln(n);
    soma:= n*(n+1)/2;
    write('O somatorio dos primeiros ');
    writeln(n:0:0 , ' numeros inteiros positivos e ', soma:0:0);
    readkey;
end.

```

Fatorial de um número n qualquer.

```

program fatorial;
uses crt;

var n,i, fat: longint; {integer?}

begin
    clrscr;
    write('Digite um numero: ');
    readln(n);
    fat:=1;
    for i:=1 to n do
    begin
        fat:=fat * i;
    end;
    writeln('O fatorial de ', n , ' e ', fat);
    readkey;
end.

```

Cálculo do valor da série E

```

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);
    READLN;
END.

```

Leitura de uma quantidade de números n e cálculo do fatorial destes n números.

```

PROGRAM EX3;
USES CRT;
VAR n, num, i, j:INTEGER;

```

```

    fat: REAL;
BEGIN
  CLRSCR;
  WRITELN('Digite a quantidade de numeros que serao lidos');
  READLN(n);
  FOR i := 1 TO n DO
  BEGIN
    WRITELN;
    WRITELN('Digite o ', i, ' º numero');
    READLN(num);
    fat := 1;
    FOR j := 1 TO num DO
    BEGIN
      fat := fat * j;
    END;
    WRITELN('Fatorial de ', num, ' = ', fat:5:2);
  END;
  READLN;
END.

```

Estrutura de Repetição com Número Indefinido de Repetições

Estrutura *Enquanto* (While – Pascal)

Utilizada quando não se conhece o número de repetições (também pode ser utilizada quando se conhece este número, mas é necessária a manipulação de uma variável contadora do laço).

O teste condicional para encerrar ou não as repetições é feito no início da estrutura, portanto, poderá haver situações onde o(s) comando(s) da estrutura não será executado.

Forma geral do comando *Enquanto*:

Enquanto **condição** Faça
Comando

Enquanto **condição** Faça
Início
Comando1
Comando2
Comando3
Fim

Os comandos são executados enquanto a **condição** for verdadeira.

Exemplo:

```

x ← 1
y ← 5
enquanto x < y faça
início
    x ← x + 2
    y ← y + 1
fim

```

Simulação deste pedaço de algoritmo:

X	y	Valores
1	5	Valores iniciais
3	6	Valores obtidos dentro da estrutura de repetição
5	7	
7	8	
9	9	

No Pascal, a correspondente a estrutura *Enquanto* é a estrutura *While*.

Forma geral:

```
while condição do
    comando;
```

```
while condição do
begin
    comando1;
    comando2;
end;
```

Estrutura de Repetição com Número Indefinido de Repetições

Estrutura *Repita* (Repeat – Pascal)

Utilizada quando não se conhece o número de repetições (também pode ser utilizada quando se conhece este número, mas é necessária a manipulação de uma variável contadora do laço).

O teste condicional para encerrar ou não as repetições é feito no final da estrutura, portanto, o(s) comando(s) da estrutura sempre serão executados pelo menos uma vez. Outra característica que diferencia a estrutura *Repita* da estrutura *Enquanto* é que a estrutura *repita* não necessita a utilização de blocos de comandos compostos (utilização de início e fim), pois a sua própria estrutura já define o início e o fim da estrutura.

Forma geral do comando *Repita*:

```
Repita
    Comandos
Até condição
```

Os comandos são executados até que a **condição** se torne verdadeira.

Exemplo:

```
x ← 1
y ← 5
repita
    x ← x + 2
    y ← y + 1
```


até $x \geq y$

Simulação deste pedaço de algoritmo:

X	y	Valores
1	5	Valores iniciais
3	6	Valores obtidos dentro da estrutura de repetição
5	7	
7	8	
9	9	

No Pascal, a correspondente a estrutura *Repita* é a estrutura *Repeat*.

Forma geral:

```
repeat  
    comandos;  
until condição;
```

Exercícios:

A conversão de graus Fahrenheit para Celsius é obtida por $c = 5/9*(f-32)$. Faça um programa que calcule e escreva uma tabela de conversão de graus Celsius para graus Fahrenheit, cujos graus variem, em Celsius, de 0 a 100, em intervalos de 0,5 graus.

Faça um algoritmo que leia uma quantidade desconhecida de números positivos. Para cada número lido o algoritmo deve escrever, o próprio número, este número ao quadrado e sua raiz quadrada. O algoritmo deve parar quando for lido um número negativo ou o zero.

Escrever um algoritmo para gerar e escrever uma tabela com os valores do seno de um ângulo A em radianos, utilizando a série de Mac-Laurin truncada com 20 termos:

$$\text{sen}(A) = \frac{A^1}{1!} - \frac{A^3}{3!} + \frac{A^5}{5!} - \frac{A^7}{7!} \dots$$

Os valores dos ângulos A devem variar de 0.0 a 6.3 de 0.1 em 0.1.

Escrever um algoritmo que lê um número não conhecido de valores, um de cada vez, e conta quantos deles estão em cada um dos intervalos [0,25], (25,50], (50,75], (75,100]. O algoritmo para quando for lido um número negativo.

Escrever um algoritmo semelhante ao anterior que calcula as médias aritméticas de cada intervalo e as escreve, juntamente com o número de valores de cada intervalo.