

**PROCEDURES E FUNÇÕES PRÉ DEFINIDAS**

Linguagem de programação PASCAL prof. Vera Alves

**FUNÇÕES ARITMÉTICAS****Abs Function**

Retorna o valor absoluto de x.

Declaração: *function* **Abs** (x);

*Obs.:* x é do tipo inteiro ou real. O resultado é do mesmo tipo de x.

Exemplo: Abs (-2.3) = 2.3

Abs (-157) = 157

**Frac Function**

Retorna a parte fracional de x.

Declaração: *function* **Frac**(x: real): real;

Exemplo: Frac (123.45) = 0.45

Frac (-123.45) = -0.45

**Int Function**

Retorna a parte inteira de x.

Declaração: *function* **Int** (x: real): real;

Exemplo: Int (123.45) = 123.0

Int (-123.45) = -123.0

**Sqr Function**

Retorna o quadrado de x.

Declaração: *function* **Sqr** (x);

*Obs.:* x é do tipo inteiro ou real. O resultado é do mesmo tipo de x.

Exemplo: Sqr (5) = 25

Sqr (2.3) = 5.29

### **Sqrt Function**

Retorna a raiz quadrada de x.

Declaração: *function* **Sqrt** (x: real): real;

Exemplo: Sqrt (25) = 5

## FUNCÕES DE TRANSFERÊNCIA

### **Chr Function**

Retorna o caracter com o valor original (valor ASCII) da expressão x.

Declaração: *function* **Chr** (x: byte): char;

Exemplo: Chr (65) = 'A'

### **Ord Function**

Retorna o valor original da expressão x.

Declaração: *function* **Ord** (x): longint;

*Obs.*: x deve ser de um tipo ordinal.

Exemplo: Ord ('A') = 65

### **Round Function**

Retorna o valor de x arredondado para o inteiro mais próximo.

Declaração: *function* **Round** (x: real): longint;

Exemplo: Round (27.8) = 28

Round (27.3) = 27

### **Trunc Function**

Retorna o valor de x truncado em inteiro.

Declaração: *function* **Trunc** (x: real): longint;

Exemplo: Trunc (27.8) = 27

Trunc (27.3) = 27

## **FUNÇÕES ORDINAIS**

### **Odd Function**

Retorna True se x é ímpar e False se x é par.

Declaração: *function* **Odd** (x: longint): boolean;

Exemplo: Odd (10) = False

Odd (5) = True

### **Pred Function**

Retorna o predecessor de x.

Declaração: *function* **Pred** (x);

*Obs.:* x deve ser de um tipo ordinal. O resultado é do mesmo tipo de x.

Exemplo: Pred ('F') = 'E'

Pred (5) = 4

### **Succ Function**

Retorna o sucessor de x.

Declaração: *function* **Succ** (x);

*Obs.:* x deve ser de um tipo ordinal. O resultado é do mesmo tipo de x.

Exemplo: Succ ('E') = 'F'

Succ (4) = 5

## PROCEDURES ORDINAIS

### **Dec Procedure**

Decrementa x.

Declaração: *procedure* **Dec** (var x [; N: longint]);

Exemplo: Dec (x); {x: = x - 1}

Dec (x,5); {x: = x - 5}

### **Inc Procedure**

Incrementa x.

Declaração: *procedure* **Inc** (var x [; N: longint]);

Exemplo: Inc (x); {x: = x + 1}

Inc (x,5); {x: = x + 5}

## OUTRAS

### **UpCase Function**

Retorna o caractere (letra) maiúscula.

Declaração: *function* **UpCase** (c: char): char;

Exemplo: UpCase ('m') = 'M'

### **Random Function**

Retorna um número randômico (aleatório).

Declaração: *function* **Random** [(Range: word)]: word; ou : real;

Exemplo: Random; {retorna valor entre [0.0, 1)}

Random (M); {retorna valor entre [0,M)}

## PROCEDURES PARA STRING

### **Delete Procedure**

Remove a quantidade de caracteres especificadas em **L** da cadeia **S**, começando com o caractere na posição **P**.

Declaração: *Procedure Delete* (S: string; P, L: integer);

### **Insert Procedure**

Inclui a cadeia **Source** na cadeia **Target** na posição **Index**.

Declaração: *Procedure Insert* (Source: string; Var Target: string; Index: integer);

### **Str Procedure**

Converte um número real ou inteiro em uma cadeia.

Declaração: *Procedure Str* (I: integer; [: length,] Var S: string);  
*Procedure Str* (R: real; [: length: decimals,] Var S: string);

### **Val Procedure**

Tenta converter **S** em um valor numérico (R ou I). Se a conversão tiver êxito, o Turbo Pascal irá atribuir a **Code** o valor zero. Se não tiver êxito, **Code** conterá um inteiro que representa a posição do caractere que causou o erro.

Declaração: *Procedure Val* (S: string; Var R: real; Var code: integer);  
*Procedure Val* (S: string; Var I: integer; Var code: integer);

## FUNCÇÕES PARA STRING

### **Copy Function**

Retorna uma parte da cadeia **S**, que inicia no número de caracter **P** e contém a quantidade de caracteres especificada em **L**.

Declaração: *Function Copy* (S: string; P, L: integer): string;

### **Concat Function**

Combina qualquer quantidade de cadeias e as retorna como uma única cadeia. Se a cadeia concatenada tiver mais de 255 caracteres, o Turbo Pascal irá gerar um erro de execução.

Declaração: *Function Concat* (S1, S2,..., Sn): string;

### **Length Function**

Retorna o tamanho da cadeia **S**.

Declaração: *Function Length* (S: string): integer;

### **Pos Function**

Retorna a posição de **SubS** em **S**. Se **SubS** não for encontrada em **S**, **Pos** retorna 0.

Declaração: *Function Pos* (SubS, S: string): integer;