

FUNCTION

```
Program AprenderFuncoes;
CONST
    SAIR = 0;
    BINARIO = 2;
    OCTAL = 8;
    HEXADECIMAL = 16;
    DECIMAL = 10;
VAR
    oper:integer;
    numero:integer;

{*
* ler
*
*     Esta função mostra um menu de comandos e lê a escolha
*     do utilizador.
*     Enquanto o utilizador não der um comando certo a função
*     insiste com o menu.
*}
FUNCTION ler:integer;
VAR
    resultado:integer;
    valor:char;
BEGIN
    resultado:=-1;
    while resultado = -1 do BEGIN
        writeln;
        writeln('Menu');

        writeln('s- sair');
        writeln('b- escrever valor em binário');
        writeln('h- escrever valor em hexadecimal');
        writeln('o- escrever valor em octal');
        writeln('d- ler outro número em decimal para converter');
        writeln;
        readln(valor);
        case valor of
            'S','s':resultado:=SAIR;
            'B','b':resultado:=BINARIO;
            'H','h':resultado:=HEXADECIMAL;
            'O','o':resultado:=OCTAL;
            'D','d':resultado:=DECIMAL;
        end;
    END;
    ler:=resultado;  {É assim que se devolve o valor da Função}
END;
```

```
{*
 * Dec_Bin
 *
 *     Esta função tem como entrada um valor em Decimal e
 *     converte esse número para Binário.
 *     A conversão é feita através das divisões sucessivas.
 *     A função é recursiva (chama-se a si mesma), porque é
 *     necessário dividir primeiro e só depois é que se podem
 *     escrever os valores na ordem inversa;
 *     A recursividade termina quando o valor a dividir for 0.
 *     Nessa altura a função retorna ao local de onde foi chamada
 *     e escreve o valor do resto;
```

```
*}
```

```
FUNCTION Dec_Bin(valor:integer):integer;
```

```
VAR
```

```
    quociente:integer;
```

```
    resto:integer;
```

```
BEGIN
```

```
    if(valor <> 0) then BEGIN
```

```
        quociente:=valor div 2;
```

```
        resto:=valor-quociente * 2;
```

```
        Dec_Bin(quociente);
```

```
        write(resto);
```

```
    END;
```

```
END;
```

```
{*
```

```
 * Dec_Oct
```

```
 *
```

```
 *     Esta função tem como entrada um valor em Decimal e
 *     converte esse número para Octal.
```

```
 *     A conversão é feita através das divisões sucessivas.
```

```
 *     A função é recursiva (chama-se a si mesma), porque é
 *     necessário dividir primeiro e só depois é que se podem
 *     escrever os valores na ordem inversa;
```

```
 *     A recursividade termina quando o valor a dividir for 0.
```

```
 *     Nessa altura a função retorna ao local de onde foi chamada
 *     e escreve o valor do resto;
```

```
*}
```

```
FUNCTION Dec_Oct(valor:integer):integer;
```

```
VAR
```

```
    quociente:integer;
```

```
    resto:integer;
```

```
BEGIN
```

```
    if(valor <> 0) then BEGIN
```

```
        quociente:=valor div 8;
```

```
        resto:=valor-quociente * 8;
```

```
        Dec_Oct(quociente);
```

```
        write(resto);
```

```
    END;
```

```

END;

{*
* Dec_Hex
*
*     Esta função tem como entrada um valor em Decimal e
*     converte esse número para Hexadecimal.
*     A conversão é feita através das divisões sucessivas.
*     A função é recursiva (chama-se a si mesma), porque é
*     necessário dividir primeiro e só depois é que se podem
*     escrever os valores na ordem inversa;
*     A recursividade termina quando o valor a dividir for 0.
*     Nessa altura a função retorna ao local de onde foi chamada
*     e escreve o valor do resto;
*}
FUNCTION Dec_Hex(valor:integer):integer;
VAR
    quociente:integer;
    resto:integer;
BEGIN
    if(valor <> 0) then BEGIN
        quociente:=valor div 16;
        resto:=valor-quociente * 16;
        Dec_Hex(quociente);
        if resto<9 then write(resto)
        else case resto of      {0 resto é um número hexadecimal }
            10: write('A');      { por isso é preciso escrever }
            11: write('B');      { os símbolos correctos. }
            12: write('C');
            13: write('D');
            14: write('E');
            15: write('F');
        end;
    END;
END;

BEGIN
    writeln(' Este programa permite converter um número ');
    writeln(' decimal (base 10) para outra base à escolha. ');
    writeln('Escreva o número: ');
    oper:=DECIMAL;
    while oper <> SAIR do BEGIN
        case oper of
            BINARIO: Dec_Bin(numero);
            OCTAL: Dec_Oct(numero);
            HEXADECIMAL: Dec_Hex(numero);
            DECIMAL: readln(numero);
        end;
        oper:=ler();
    END;
END.

```

Este programa tem diversas Funções. Escreve-o e põe-o a correr.

Procura compreender o que faz na globalidade e o que é que cada Função faz em particular.

O PASCAL chama de procedimentos (PROCEDURE) àquelas Funções que não precisam de devolver um valor. Ou seja, o Random é uma FUNCTION, enquanto que o Randomize, Readln e Writeln são PROCEDURE.

Para que possas programar futuramente noutras linguagens, assume que tudo são Funções, porque um procedimento é uma Função que não devolve qualquer valor.

Se olhares com atenção apercebes-te de que uma Função é um programa, ou melhor, o Programa (PROGRAM) é uma Função, a Função principal.

Por vezes, para simplificar os programas, é preciso chamar uma Função em que essa Função se chama a si mesma. Ou seja, é recursiva. Parece complicado mas não é; é como se descesses uma escada, e em cada degrau colocas as operações que fazes dentro da função. Se chamas novamente a função, descas mais um degrau e comesças tudo de novo. E vais descendo, descendo, descendo ... Depois, quando a Função sai pela primeira vez, sobes o degrau em que estás com o resultado dessa Função, e utiliza-lo aí, no degrau de cima. É provável que de seguida voltes a ter novo resultado e subas mais um degrau. E vais subindo, subindo, subindo ...

Até que chegas outra vez ao lugar de partida!

Vês, afinal programar é bem divertido ;)