

## Trabalho Prático Nº 5

Suporte à programação estruturada:

- Subrotinas (procedimentos e funções)
- Passagem de parâmetros por registo
- Passagem de parâmetros por variáveis globais
- Passagem de parâmetros por pilha
- Variáveis locais

### Passagem de parâmetros por registos

1. Implemente um procedimento que determine o valor não negativo de 16 bits representado numa cadeia de caracteres constituída por dígitos decimais.

Parâmetros:

- Endereço da cadeia de caracteres passado nos registos ES:[SI]
- Valor não negativo devolvido no registo AX

2. Escreva um procedimento que implemente o processo de conversão inverso ao referido no exercício 1.

3. Escreva um procedimento que, sendo dado:

- o endereço do primeiro elemento de uma tabela de valores do tipo WORD nos registos ES: [SI];
- o número de elementos na tabela no registo AX;

ordene a tabela por ordem decrescente.

### Passagem de parâmetros por variáveis globais

4. Implemente um procedimento que determine o valor de 16 bits representado numa cadeia de caracteres constituída por dígitos hexadecimais.

Parâmetros:

- Variável global Pt = Ponteiro para a variável do tipo cadeia de caracteres
- Variável global N = Valor de 16 bits

5. Implemente um procedimento que efectue o processo de conversão inverso ao referido no exercício 4, ou seja, sendo dado um valor de 16 bits, represente este valor numa cadeia de caracteres de dígitos hexadecimais.

### Passagem de parâmetros por pilha

**Para os exercícios seguintes, deve apresentar a organização da pilha do sistema (“stack frame”) aquando da chamada do respectivo subprograma.**

6. Considere uma tabela de valores do tipo BYTE que correspondem a valores das N amostras de um sinal eléctrico quantificado digitalmente. Escreva um procedimento que processe os valores da tabela de acordo com:

$$\begin{cases} Y[k] = X[k] & \text{com } k = 1, 2, N \\ Y[k] = \frac{1}{4} \sum_{i=-1}^2 X[k-i] & \text{com } k \in \{3, \dots, (N-1)\} \end{cases}$$

em que  $X[k]$  representa a amostra  $k$  da sequência de amostras numeradas de  $1..N$  e  $Y[k]$  representa os valores da tabela depois de processados.

Escreva o procedimento de acordo com o seguinte cabeçalho:

```
Const N=10;
Type TTab=Array [1..N] of byte;
Procedure Processa (Var X:TTab; N:Word);Far;Assembler;
```

7. Escreva um procedimento que determine o valor de uma potência de acordo com o seguinte cabeçalho:

```
Procedure Potencia (base, expoente:Word;Var pot:Word);
Near;Assembler;
```

8. Implemente um procedimento que modifique uma cadeia de caracteres, retirando todos os caracteres que sejam uma repetição do caractere <espaço>. Por exemplo, a cadeia de caracteres

“Os alunos de Engenharia dos Sistemas e Informática”  
deve ser transformada para  
“Os alunos de Engenharia dos Sistemas e Informática”

```
Procedure RetiraEsp (Var s:String);Far; Assembler;
```

9. Implemente um procedimento que ordene uma cadeia de caracteres por ordem alfabética.

Por exemplo, a cadeia de caracteres  
“Engenharia”  
deve ser transformada para  
“Eaaeghinnr”

```
Procedure OrdenaStr (Var s:String); Near; Assembler;
```

10. Escreva um subprograma que transforme em maiúscula a primeira letra de cada palavra armazenada numa cadeia de caracteres.

```
Procedure CvtMaius (s:string); Near; Assembler;
```

11. Implemente um procedimento que troque o valor de duas variáveis do tipo integer.

```
Procedure Troca (Var a,b:Integer);Far; Assembler;
```

12. Considere as seguintes declarações:

```
Const N=10;
Type Data=Record
    Dia,Mes:Byte;
    Ano:Word;
End;
TData=Array [1..N] of Data;
Var TD: TData;
```

Os campos do registo Data permitem a representação de uma data. Escreva um procedimento que determine a maior data armazenada na tabela TD.

```
Procedure MaxData (Var TD:TData;N:Word;Var D:Data);Far;
Assembler;
```

13. Escreva uma função que determine o máximo de dois valores.

```
Function Maximo(a,b:Integer):Integer;Near;Assembler;
```

14. Escreva uma função que calcule o número de ocorrências de um determinado carácter numa variável do tipo string.

```
Function ContaChar(s:String;c:Char):Byte; Near; Assembler;
```

15. Escreva uma função que determine o n-ésimo elemento da sequência de Fibonacci, definida por:

$$\begin{cases} U_N = N & , N = 0,1 \\ U_N = U_{N-1} + U_{N-2} & , N \geq 2 \end{cases}$$

```
Function Fibonacci(N:Byte):byte; Near; Assembler;
```

16. Escreva uma função que determine o valor binário de 8 bits representado numa cadeia de caracteres constituída por '0' e '1'.

```
Function CvtBinario(s:String):Byte; Near; Assembler;
```