UNISINOS - UNIVERSIDADE DO VALE DO RIO DOS SINOS
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS (C6/6) – Curso: Informática

LABORATÓRIO I – Programas com Sub-Rotinas: Procedure / Function

Disciplina: Linguagem de Programação PASCAL **E-mail:** ***osorio@exatas.unisinos.br***
Professor responsável: ***Fernando Santos Osório*** **Web:**
Semestre: 2000/2 ***http://www.inf.unisinos.br/~osorio/lab1.html***
Horário: 63 **Xerox : *Pasta 54* – LAB. I (Xerox do C6/6)**

Lista de Exercícios - Lista número: 05 - Programas com Sub-Rotinas

⇒ *Funções e Procedimentos – FUNCTION / PROCEDURE*
Parte I - Passagem de Parâmetros por Valor

1. Faça um programa com uma sub-rotina (função) que receba 3 valores de entrada e retorne o maior valor entre estes três valores.

2. Faça um programa com uma sub-rotina (função) que calcule X elevado à $Y = X^Y$. Leia 2 valores de X e Y e exiba o resultado da chamada da sub-rotina na tela. Exemplo: 2 elevado à 3 é igual à 2*2*2 = 8.

3. Faça um programa que peça para ler 2 notas e depois mostre:
   - A média aritmética simples;
   - A média ponderada entre os dois valores (nota 1 com peso 1 e nota 2 com peso 2);
   - O valor necessário para recuperar a pior nota e passar com média igual ou superior a 6.0, considerando o uso da média aritmética simples (notas com pesos iguais);
   - O valor necessário para recuperar a pior nota e passar com média igual ou superior a 6.0, considerando o uso da média ponderada (nota 1 => peso 1, nota 2 => peso 2);
   - Qual das duas médias (aritmética, ponderada ou tanto faz) é mais benéfica para o aluno, ou seja, qual das duas médias deixa este aluno com a melhor nota.

   Faça este programa de forma modular, ou seja, usando uma sub-rotina (função) para o cálculo de cada tarefa descrita acima.

4. Faça uma sub-rotina “Br_UpCase” que dado um caracter qualquer retorne o mesmo caracter sempre em maiúsculo, aceitando inclusive os caracteres acentuados da língua portuguesa (por exemplo: á, é, í, ó, ú, ç, ã, õ, â, ô, à, ü). Faça um programa que leia uma palavra (string) e chame esta sub-rotina para cada um dos caracteres desta palavra, exibindo o resultado após a conversão para maiúsculo. Dicas:

```
Letra := Texto[1];  { Copia o caracter número 1 da string “Texto” para a variável tipo char “Letra” }
Texto[5] := Letra;  { Copia o conteúdo da variável tipo char “Letra” para o quinto caracter da string
                      armazenada na variável “Texto” }
```

5. Transforme o programa anterior de conversão de strings com caracteres em minúsculo para strings com caracteres em maiúsculo, em uma função. Esta função recebe uma string de entrada e devolve a string convertida para letras maiúsculas.

6. Criar um procedimento (*procedure*) que desenhe uma moldura ao redor da tela do micro (quadrado 24x80). Faça no programa principal uma chamada a esta *procedure*, desenhando a moldura e após escrevendo "Hello World" no meio da tela (Coluna X=35, Linha Y=12). Para desenhar a moldura use os caracteres especiais da tabela ASCII estendida do Turbo Pascal. Exemplo: pressione a tecla ALT e ao mesmo tempo um dos seguintes números no teclado numérico **ALT 200, ALT 201, ALT 205, ALT 186, ALT 187, ALT 188**. Exemplo:

HELLO WORLD

7. Altere o programa anterior de maneira que quando o usuário apertar uma tecla (*readkey*), a tela seja limpada, a moldura desenhada novamente e seja escrito "Bye-Bye World" no meio da tela. O programa deve terminar automaticamente após uma espera (*delay)* de 5 segundos.

8. Faça novamente um programa para o cálculo do fatorial, mas desta vez crie uma rotina separada que realize o cálculo do fatorial de um número. Utilize também a rotina que desenha uma moldura na tela ao apresentar o resultado (reaproveitamento de código).

9. Faça uma rotina genérica para criar molduras na tela com o tamanho especificado pelo programa através dos parâmetros que são passados para esta rotina. Os parâmetros vão indicar a linha inicial e final da moldura na tela, assim como a coluna inicial e final da moldura. No programa principal use esta rotina para emoldurar o seu nome escrito no centro da tela do computador.

10. Escrever um programa que obtenha a data atual (*getdate*) e imprima ela no formato textual por extenso. A escrita da data por extenso deve ser realizada por um procedimento separado. Exemplo: Data: ***01/01/2000*** => Imprimir: **Sábado, 1 de janeiro de 2000**.

11. Modificar o programa anterior para que seja lido o nome de uma pessoa e a data de seu nascimento. Em seguida o programa deve produzir uma mensagem como a que segue logo abaixo. A "data de hoje" é obtida com o comando *getdate*.

    Nome: Fulano da Silva
    Data de nascimento: 04/05/1998
    Data de hoje: 12/05/1966

    O Fulano da Silva, nascido na Segunda-Feira, 04 de maio de 1998 foi registrado neste cartório na data de hoje, Terça-Feira, 12 de maio de 1998.

12. Usando o programa que você havia feito para a validação da data, transforme este programa numa função que retorna um valor indicando se a data é válida ou não. Altere o programa anterior de maneira que este valide a data fornecida pelo usuário. Se a data for inválida, leia novamente a data de nascimento, valide e repita o processo tantas vezes quantas forem necessárias, até que o usuário forneça uma data correta.

⇒ *Funções e Procedimentos – FUNCTION / PROCEDURE*
Parte II - Passagem de Parâmetros, Variáveis Locais e Globais
"Teste de Mesa"

1. Procure *prever o comportamento do programa* abaixo e depois teste o programa e indique qual é o valor das variáveis A, B e C que é escrito na tela – dentro da procedure e no final do programa.

```
Program Teste_de_Mesa;

Procedure  Altera (A, B : Integer );
Var
   C:Integer;
Begin
    Writeln ('Dentro da Altera - Início: A,B,C = ',A,' - ',B,' - ',C);
    A := 10;
    B := 10;
    C := 10;
    WriteLn ('Dentro da Altera - Fim: A,B,C = ',A,' - ',B,' - ',C);
End;

Var
    A, B, C: Integer;
Begin
    A := 1;
    B := 2;
    C := 3;
    WriteLn ('Antes da Altera: A,B,C = ',A,' - ',B,' - ',C);
    Altera (B, A);
    WriteLn ('Depois da Altera: A,B,C = ',A,' - ',B,' - ',C);
    ReadLn;
End.
```

Variável	A	B	C
Antes da chamada da procedure Altera			
Dentro da procedure Altera – Início			
Dentro da procedure Altera – Fim			
Depois da chamada da procedure Altera			

2. Execute passo-à-passo o programa anterior que possui uma variável local declarada dentro da procedure (variável C). Coloque um "watch" nesta variável e observe o seu valor (execute o passo-à-passo com uso da tecla F7). O que ocorre com a variável?

3. Execute passo-à-passo o programa anterior que possui duas variáveis passadas como parâmetro **por valor** para a procedure (variáveis A e B). Coloque um "watch" nestas variáveis e observe o seu valor. O que ocorreu com as variáveis originais ao voltar da execução da procedure? (Quando as variáveis originais não são afetadas, isto significa que estamos usando um tipo de *passagem de parâmetros por valor*).

4. Altere o programa acima, invertendo a ordem da declaração dos parâmetros A e B, ficando a declaração da procedure assim: Procedure Altera (**B, A** : Integer);
   Execute passo-à-passo o programa observando o efeito da ordem dos parâmetros nos valores de A e B dentro da procedure.

5. Altere o programa original acima, colocando a palavra VAR (passagem de parâmetros por referência) na frente dos parâmetros A e B, ficando assim a declaração da procedure:
   Procedure Altera (**VAR** A, B: Integer); => Execute passo-à-passo e veja o resultado!
   O que acontece neste caso com o 'var' se você chamar a 'altera' assim: altera(1,2); ?

⇒ *Funções e Procedimentos – FUNCTION / PROCEDURE*
Parte III - Passagem de Parâmetros por Referência

1. Faça um programa que leia 2 valores e chame uma sub-rotina (procedure) que receba estas 2 variáveis e troque o seu conteúdo, ou seja, esta rotina é chamada passando duas variáveis A e B, por exemplo, e após a execução da rotina A conterá o valor de B e B terá o valor de A.

2. Altere o programa anterior de forma a ler 4 valores em 4 variáveis: A, B, C, D. Use a rotina "troca_valores" implementada no programa anterior para trocar os valores de A com B, depois de C com D, mostre como ficaram os valores das variáveis, e para concluir destroque seus conteúdos mostrando novamente na tela como ficaram estas variáveis.

3. Faça um programa que leia dois valores e chame uma sub-rotina que receba estes 2 valores de entrada e retorne o maior valor na primeira variável e o segundo maior valor na segunda variável. Escreva o conteúdo das 2 variáveis na tela.

4. Faça um programa que leia três valores e chame uma sub-rotina que receba estes 3 valores de entrada e retorne o maior valor entre estes três valores na primeira variável, o segundo maior valor na segunda variável e o terceiro maior valor na terceira variável. Exibir os valores ordenados na tela (sugestão: use a sub-rotina "troca_valores" do exercício 1).

5. Faça um programa que leia um número e gere todos os números primos entre 1 e este número fornecido, exibindo-os na tela. O programa deve ter uma sub-rotina que determine se um determinado número é ou não primo.

6. Usando a sub-rotina que exibe uma moldura na tela em uma janela (região) definida pelo usuário, faça uma outra sub-rotina que escreva uma mensagem na tela pedindo para ler 2 números, leia estes números e retorne para o programa principal. A mensagem e a leitura dos 2 números deve ser feita dentro de uma janela com moldura, sendo todo este procedimento implementado por uma sub-rotina (ler_numeros). Em seguida exiba a media aritmética simples destes dois números também usando uma janela com moldura. Exemplo:

   Entre com 2 números: ***5.2 7.4***

   Média: **6.3**

7. Faça um programa que leia 5 números. Este programa deve ter um menu que permita ao usuário escolher qual opção de cálculo ele deseja realizar: média aritmética simples, média ponderada (ler os pesos associados a cada nota que serão informados pelo usuário), desvio padrão, maior valor e menor valor. A leitura dos 5 valores também deve ser uma das opções do menu.

8. Faça um jogo no qual o computador desafia dois usuários para ver quem tem mais memória. O jogo deve gerar um número de 0 à 9 e apresentar ao jogador 1 durante 5 segundos. Depois deve repetir este procedimento para o jogador 2. Em seguida, ele deve pedir ao jogador 1 para informar o número escolhido, e após pedir ao jogador 2 para fazer a mesma coisa. Na segunda rodada, o computador deve gerar mais um número entre 0 e 9, que será "concatenado" à direita do primeiro número (string). Então repetiremos o procedimento de exibição e teste de memorização para os dois usuário. O programa deve continuar adicionando números (casas adicionadas ao final do número) até que um dos dois jogadores não consiga mais memorizar perfeitamente o número. Para concluir, o programa deve informar qual dos dois jogadores ganhou, quantas casas foram memorizadas ao total, e em que casa foi que o jogador perdedor errou (primeira casa errada começando da esquerda para a direita). Dicas: procure desenvolver o programa em módulos; armazene a seqüência de números dentro de uma string. Seu programa ficou bem modular? Seria fácil de adaptá-lo para 3 jogadores?