

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

LABORATÓRIO II – AULA 03

Disciplina: Linguagem de Programação PASCAL
Professor responsável: Fernando Santos Osório
Semestre: 99/2
Horário: 21 e 41

E-mail: osorio@exatas.unisinos.br
Web: <http://www.inf.unisinos.br/~osorio/lab2.html>
Xerox : Pasta 54 – LAB. II (Xerox do C6/6)

EXERCÍCIOS – Programas Recursivos

1. Faça um programa recursivo para calcular o fatorial de um número. Lembre-se de usar uma variável com uma maior precisão (longint) a fim de reduzir o problema referente a manipulação de números com valores muito grandes. A definição formal do cálculo do fatorial é a seguinte:

Fatorial (X) => (a) 1 se $X = 0$;
(b) $X * \text{Fatorial}(X-1)$ se $X > 0$;

2. Faça uma função recursiva para calcular o Máximo Divisor Comum (MDC) entre dois números. Uma forma de se calcular o MDC entre dois número x e y é:

MDC (X, Y) => (a) Y se $Y \leq X$ e $X \bmod Y = 0$
(b) MDC (Y, X) se $X < Y$
(c) MDC (Y, X mod Y) se caso contrário

3. Faça uma função recursiva para calcular a soma dos elementos de uma matriz. Dica: use a seguinte definição para a função:

Function SomaMatriz(V:Matriz;LinInicial,LinFinal,ColInicial,ColFinal:integer):integer;

4. Faça uma função recursiva para pesquisar um determinado número em um vetor ordenado. A função deve receber o vetor de inteiros (já ordenado) como parâmetro, juntamente com o início e fim, e um número que deverá ser procurado. Desenvolva a função de modo que ela use o seguinte algoritmo para achar o número:

1. Pegue o elemento central (entre o início e o fim) e compare com o número a ser procurado;
2. Se o número for menor que o elemento central pesquise nos elementos entre o início e o elemento central, caso contrario pesquise nos elementos entre o elemento central e o fim.
3. Caso o número não esteja no vetor retorne -1, senão retorne a posição onde ele foi encontrado.

Este algoritmo é conhecido como Pesquisa Binária.

5. Implemente um algoritmo do tipo “Flood-Fill” para o preenchimento de áreas em desenhos. Este programa deve receber como entrada um arquivo texto contendo uma matriz representando a imagem (64x64 caracteres). Dada uma coordenada nesta matriz, realizar o preenchimento (função de derramar um balde de tinta de softwares como o PaintShop), sendo que o preenchimento deve se limitar as bordas estabelecidas no desenho. Salvar em disco um arquivo texto contendo a imagem após o seu preenchimento. A figura abaixo demonstra um exemplo da aplicação deste algoritmo:

- Imagem 64x64: 64 caracteres em cada linha – 64 linhas com 64 caracteres cada;
- Exemplo do conteúdo do arquivo texto, considerando-se uma imagem de apenas 8x8;

```

1 2 3 4 5 6 7 8 #
. . . . . . . . 1
. . * * * * * . 2
. . * . . . * . 3
. . * . . . * . 4
. . * . . . * . 5
. . * * * . * . 6
. . * . $ . * . 7
. . * * * * * . 8

```

Comando: Flood_Fill (Matriz_Imagem, 4, 5, ‘*’, ‘.’);

Parâmetros: Matriz_Imagem = Matriz contendo os caracteres que definem a imagem
 4,5 = Coordenada do ponto inicial onde será derramada a tinta (lin, col).
 ‘*’ = Tipo de tinta para preencher (‘*’ simboliza uma cor específica)
 ‘.’ = Tipo de tinta do fundo a ser preenchido

Antes

```

. . . . . . . .
. . * * * * * .
. . * . . . * .
. . * . . . * .
. . * . . . * .
. . * * * . * .
. . * . $ . * .
. . * * * * * .

```

Depois

```

. . . . . . . .
. . * * * * * .
. . * * * * * .
. . * * * * * .
. . * * * * * .
. . * * * * * .
. . * . $ * * .
. . * * * * * .

```

Comandos de manipulação em modo gráfico do Turbo Pascal 7: (para os alunos que queiram implementar uma versão gráfica deste exercício)

- InitGraph , GetPixel, PutPixel, Line, Rectangle, Circle, SetColor (ver o help do TP7).