

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

LABORATÓRIO II – AULA 05

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 – Pilhas com alocação estática

1. Faça um programa para a manipulação de pilhas com alocação sequencial e estática na memória do computador. Crie as seguintes rotinas genéricas de manipulação de dados:

Definições elementares:

Const

Tamanho = 10;

Type

Tipo_Dado = Integer;

Tipo_Pilha = Record

Dado: array [1..Tamanho] of Tipo_Dado;

Base,Topo: Integer;

End;

Rotinas:

Procedure Inicializa_Pilha (Var P: Tipo_Pilha);

Function Insere_Pilha (Var P: Tipo_Pilha; Dado: Tipo_Dado): Boolean;

Function Retira_Pilha (Var P: Tipo_Pilha; Var Dado: Tipo_Dado): Boolean;

Function Pilha_Vazia (P: Tipo_Pilha): Boolean; { Testa para ver se a pilha está vazia }

Function Quantidade_Pilha (P: Tipo_Pilha): Integer; { Indica quantos dados tem na pilha }

Procedure Lista_Pilha (Var P: Tipo_Pilha); { Desempilha para listar os dados }

Procedure Exibe_Pilha (P: Tipo_Pilha); { Exibe os dados, mantendo a pilha inalterada }

Procedure Inverte_Pilha (Var P: Tipo_Pilha); { Inverte a ordem dos dados da pilha }

Procedure Esvazia_Pilha (Var P: Tipo_Pilha); { Desempilha todos os dados da pilha }

EXERCÍCIOS – Filas com alocação estática

1. Faça um programa para a manipulação de filas com alocação seqüencial e estática na memória do computador. Crie as seguintes rotinas genéricas de manipulação de dados:

Definições elementares:

```

Const
    Tamanho = 10;
Type
    Tipo_Dado = Integer;
    Tipo_Fila = Record
        Dado: array [1..Tamanho] of Tipo_Dado;
        Inicio, Fim: Integer;
    End;

```

Rotinas:

```

Procedure Inicializa_Fila (Var F: Tipo_Fila);

Function  Insere_Fila (Var F: Tipo_Fila; Dado: Tipo_Dado): Boolean;

Function  Retira_Fila (Var F: Tipo_Fila; Var Dado: Tipo_Dado): Boolean;

Function  Fila_Vazia (F: Tipo_Fila): Boolean; { Testa para ver se a fila está vazia }

Function  Quantidade_Fila (F: Tipo_Fila): Integer; { Indica quantos dados tem na fila }

Procedure Lista_Fila (Var F: Tipo_Fila); { Retira os dados da fila para mostra-los }

Procedure Exibe_Fila (F: Tipo_Fila); { Exibe os dados, mantendo a fila inalterada }

Procedure Inverte_Fila (Var F: Tipo_Fila); { Inverte a ordem dos dados da fila }

Procedure Esvazia_Fila (Var F: Tipo_Fila); { Retira todos os dados contidos na fila }

```

2. Faça um programa para simular as filas de um banco (considere 5 filas de atendimento). O laço do programa principal deve conter os seguintes comandos:
 - Sortear (random) uma das filas e adicionar um cliente novo que chegou ao banco;
 - Sortear (random) uma das filas para ser atendido um cliente. Retirar um cliente da fila que foi sorteada para ser atendida;
 - Mostrar na tela o estado das filas, ou seja, a quantidade de clientes que tem em cada uma das filas;
 - Esperar 1 segundo (delay);
 - Repetir este procedimento 20 vezes.

Obs: Sorteio => Qual_Fila := random(5)+1; / Espera => delay (1000);
 Use o comando “randomize” no início do programa.