

Universidade Estadual de Maringá
Departamento de Informática
Disciplina 1207 – Algoritmos e Estruturas de Dados I – Turma 31 - 2007
Profa. Josiane

TRABALHO 3º BIMESTRE

Instruções gerais:

- 1) Os exercícios abaixo devem ser implementados em linguagem Pascal;
- 2) Somente os arquivos .pas deverão ser entregues. Coloquem nomes que representem o exercício nos arquivos. Exemplo: exe01.pas ou goldbach.pas
- 3) Serão avaliados:
 - a. A corretude do programa em relação ao que foi pedido no exercício;
 - b. A colocação em prática dos conceitos que foram discutidos em sala de aula de forma correta;
 - c. A **eficiência** do algoritmo implementado;
 - d. O capricho da implementação e a endentação;
 - e. **A estruturação do programa e módulos e a nomeação das variáveis, funções e procedimentos;**
 - f. A forma de interação com o usuário.
- 4) Todos os exercícios devem ser compactados juntos (zipados) e enviados por e-mail para josianempf@gmail.com, com o subject “Trab3AED” (sem espaços), até as 23:59hs do dia 12/09/2007. Coloque seu nome e R.A. como nome do arquivo compactado. Exemplo: maria42536.zip (favor utilizar esta forma de compactação);
- 5) **Não** serão avaliados os trabalhos:
 - a. Que cheguem fora do prazo;
 - b. Que não compilem;
 - c. Que não foram compactados em um só arquivo;
 - d. Que não tiverem identificação (nome e R.A.);
 - e. Que não seguirem todas estas instruções;
 - f. Que possuírem código semelhante.
- 6) Não se esqueça que o trabalho vale 20% da nota do bimestre, e que é uma preparação para a prova do bimestre.

Exercícios do trabalho: Todos os programas devem ser feitos utilizando funções e procedimentos de forma adequada. Comente qual é o objetivo de cada um desses módulos.

- 1) Suponha que duas pilhas P1 e P2 poderão ter no máximo 50 elementos cada uma, mas a qualquer momento as duas pilhas juntas não poderão ter mais do que 70 elementos. Assim, ao invés de dois vetores de 50 elementos, podemos aloca-las em um vetor de 70 elementos. Implemente um programa que crie esta estrutura de dados e permita as operações de inserção, remoção e impressão das pilhas.
- 2) Um deque pode ser visto como uma pilha/fila na qual se pode inserir e remover nas duas extremidades. Podemos inserir um elemento na esquerda e retirar um elemento

da direita e vice-versa. Implemente as funções que manipulam um deque dinâmico (com ponteiros) e faça um programa que permita ao usuário:

- (a) Inserir um elemento na esquerda;
- (b) Inserir um elemento na direita;
- (c) Remover elemento da esquerda;
- (d) Remover elemento da direita;
- (e) Visualizar os elementos do deque;
- (f) Verificar se o deque está vazio.

3) Implemente uma Unit em Pascal que represente um Tipo Abstrato de Dados Pilha (dinâmico). Utilize este TAD para implementar um programa que receba uma expressão em notação infixa. Exemplo: $[(32*5) + (55/7) - 1]$. Considere apenas expressões com número inteiro, mas que podem resultar em um número real. O programa deve:

- (a) Transformar a expressão da notação infixa para a notação polonesa reversa utilizando uma pilha. Exemplo: $32\ 5\ *\ 55\ 7\ /\ +\ 1\ -$ é a expressão acima em notação polonesa reversa.
- (b) Resolver a expressão polonesa reversa utilizando a mesma pilha e mostrar o resultado final da expressão. Exemplo: 166.857 seria o resultado da expressão acima.

Para resolver a expressão basta empilhar os operandos e quando encontrar um operador, desempilhar os operandos (2) e fazer a operação. O resultado deve ser novamente empilhado.

Dica: utilize o procedimento **Val** do Pascal se precisar fazer conversão de char ou string para inteiro. (o procedimento **Str** faz o contrário).

4) Suponha arquivos textos nos quais em cada linha está armazenado o R.A. de uma aluno e em seguida sua nota em determinada disciplina. Por exemplo:

42554	7,3	Um professor tem estes arquivos armazenados, um por grupo de trabalho dos alunos de uma mesma turma. O professor deseja agora que estes dados sejam armazenados todos somente em um arquivo organizado por ordem de R.A. (do menor para o maior). Implemente um programa que permita ao professor:
41557	8,2	
41785	3,2	
41585	2	
41569	9,3	

- (a) Ler um arquivo de dados determinado pelo professor. (Coloque os dados em uma lista dinâmica para permitir outras operações – Implemente um TAD de uma lista dinâmica para esta estrutura);
- (b) Visualizar o conteúdo de um arquivo;
- (c) Ordenar os dados de um arquivo (por R.A.), depois de tê-lo lido. (Insira elemento por elemento na lista conforme a leitura é feita de forma que eles fiquem ordenados);
- (d) Visualizar os dados de um arquivo lido de forma ordenada;
- (e) Juntar o conteúdo de dois arquivos já lidos e ordenados. O professor pode ter no máximo dois arquivos lidos em determinado momento. (Implemente uma operação que receba duas listas ordenadas, e junte as duas listas em uma terceira

também ordenada (chamada de **merge**). Você deve percorrer as duas listas de forma ordenada – do início para o fim - e inserir os elementos **sempre no fim** da terceira lista – não é permitida utilizar uma operação que já insere os elementos na ordem para esta operação);

(f) Escrever o conteúdo ordenado em um outro arquivo cujo nome é fornecido pelo professor.

- 5) Um gerente de um banco está interessado em saber se o atendimento nos guichês de caixa se torna mais rápido utilizando filas individuais para cada guichê ou utilizando uma fila única de atendimento. O banco possui 5 guichês de caixa. Em cada unidade de tempo (laço de repetição) existe a possibilidade de até 6 pessoas chegarem ao banco e se colocarem na fila (ou filas). As pessoas que chegam entram sempre na fila menor. (se houve mais que uma). Cada pessoa leva até 10 unidades de tempo no guichê do caixa (sendo atendida). Quando o atendimento de uma pessoa termina, outra imediatamente é atendida. A cada 5 unidades de tempo existe a possibilidade de que um idoso, deficiente ou gestante chegue na agência. Uma fila de espera especial deve ser feita para estes clientes com prioridade de atendimento. Quando um guichê de caixa desocupar, esta fila terá sempre prioridade. Seu trabalho é simular esta situação e fornecer informações ao gerente para que ele possa tomar sua decisão.

Para isso comece implementando uma Unit em Pascal que represente um Tipo Abstrato de Dados Fila (dinâmico). Utilize este TAD para implementar um programa que simule o atendimento bancário considerando as duas possibilidades de atendimento:

- (a) Com apenas uma fila de espera; e
- (b) Com uma fila de espera para cada guichê (5 no caso).

A cada **5** unidades de tempo seu programa deve fornecer ao gerente da agência as seguintes informações para **cada** uma das possibilidades de atendimento:

- (a) A quantidade de pessoas esperando na fila (ou filas);
- (b) O tempo médio de **espera** (sem contar o tempo de atendimento) dos clientes que já foram atendidos;
- (c) O tempo médio de **atendimento** dos clientes que já foram atendidos;
- (d) A quantidade total de clientes atendidos até o momento.