

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

TRABALHO 2º 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 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 “Trab2AED” (sem espaços), até as 23:59hs do dia 20/06/2007. Coloque seu nome e R.A. como nome do arquivo compactado. Exemplo: maria42536.zip;
- 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, o que pode ser a diferença entre aprovação e exame.

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) Faça um programa que seja capaz de encontrar uma determinada palavra dentro de uma matriz (N x N). Declare uma matriz N x N (N linhas e N colunas – use constantes) - essa matriz será utilizada como um caça-palavras. Em cada posição da matriz haverá 1 caractere. A matriz deverá ser lida no início do programa. Então, o programa deverá permitir que o usuário entre com uma palavra, e o programa deverá informar se ela foi ou não encontrada na matriz. O usuário deve ter a opção de continuar procurando por outras palavras, até que ele decida parar. O Algoritmo deverá ser capaz de encontrar qualquer palavra informada pelo usuário, analisando uma matriz NxN contendo quaisquer elementos.

Exemplo:

E	A	Q	T	G	J	N	M	K	I
A	L	G	O	R	I	T	M	O	N
A	Z	X	C	K	X	W	E	P	T
B	D	F	Y	L	P	H	U	L	E
G	E	V	H	C	V	B	H	E	R
I	J	G	B	F	X	Z	J	I	N
J	H	N	I	C	V	O	G	U	E
K	P	M	M	N	H	O	I	A	T
Q	W	E	A	F	E	Y	D	R	L
K	J	A	E	L	I	H	W	D	F

Considere que uma palavra pode aparecer nas seguintes direções: Horizontal: (ALGORITMO); Horizontal invertida: (WHILE); Vertical: (INTERNET); Vertical invertida (UEM); Diagonal para cima à direita: (HOJE); Diagonal para cima à esquerda: (FRIO); Diagonal para baixo à direita: (BEGIN) e Diagonal para baixo à esquerda (FIM).

- 2) Faça um programa que implemente um jogo tipo 'batalha naval'. Onde o oceano é uma matriz (N x N) e os navios são seqüências de três posições da matriz (na horizontal ou na vertical). Ao iniciar o programa são gerados aleatoriamente três seqüências (gere dois números aleatórios i e j e marque as posições [i, j-1], [i, j] e [i, j+1] ou [i-1, j], [i, j] e [i+1, j] como sendo um navio. Cuidado quando i e j estiverem na borda da matriz). O desafio do usuário é adivinhar em quais posições estão os navios e afundá-los (acertando 9 palpites). Ele terá quantas tentativas precisar, mas se errar mais de 3 palpites seguidos, ele perde o jogo. A cada tentativa do usuário o programa deverá dizer se o palpite está correto ou não. Quando o usuário conseguir afundar um navio, o programa deve informar isso, dizendo quantos navios ainda faltam. Exemplo:

	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	
<i>0</i>	0	0	0	0	0	0	0	0	0	0	0
<i>0</i>	0	1	1	1	0	0	0	0	0	0	1
<i>0</i>	0	0	0	0	0	0	0	0	0	0	2
<i>0</i>	0	0	0	0	0	2	0	0	0	0	3
<i>0</i>	0	0	0	0	0	2	0	0	0	0	4
<i>0</i>	0	0	0	0	0	2	0	0	0	0	5
<i>0</i>	0	0	0	0	0	0	0	0	0	0	6
<i>0</i>	0	0	0	0	0	0	0	0	0	0	7
<i>0</i>	0	0	0	0	0	0	0	0	0	0	8
<i>0</i>	0	0	0	0	0	0	0	3	3	3	9

Como o programa deve funcionar:

<p><i>Batalha Naval</i> <i>Jogo Inicializado com sucesso!</i> <i>Atenção: linhas e colunas variam de 0 a 9</i> <i>Entre com o palpite (linha e coluna): 0 0</i> <i>Palpite errado!</i> <i>Entre com o palpite (linha e coluna): 1 2</i> <i>Palpite correto!</i> <i>Entre com o palpite (linha e coluna): 1 1</i> <i>Palpite errado!</i> <i>Entre com o palpite (linha e coluna): 1 3</i> <i>Palpite correto!</i> <i>Entre com o palpite (linha e coluna): 1 4</i> <i>Palpite correto!</i> <i>Parabéns! Você acabou de afundar um navio! Ainda faltam 2 navios.</i> <i>Entre com o palpite (linha e coluna): 1 6</i> <i>Palpite errado!</i> ... <i>Batalha Naval</i></p>	<p><i>Jogo Inicializado com sucesso!</i> <i>Atenção: linhas e colunas variam de 0 a 9</i> <i>Entre com o palpite (linha e coluna): 0 0</i> <i>Palpite errado!</i> <i>Entre com o palpite (linha e coluna): 1 2</i> <i>Palpite correto!</i> <i>Entre com o palpite (linha e coluna): 0 2</i> <i>Palpite errado!</i> <i>Entre com o palpite (linha e coluna): 2 2</i> <i>Palpite errado!</i> <i>Entre com o palpite (linha e coluna): 1 1</i> <i>Palpite errado!</i> <i>Entre com o palpite (linha e coluna): 0 1</i> <i>Palpite errado!</i> <i>Fim de Jogo! Você perdeu!</i></p>
--	---

- 3) Faça um programa que leia uma matriz A [X, Y] (X linhas e Y colunas) e uma matriz B [Y, Z] (Y linhas e Z colunas) de números inteiros, mostre-as, e em seguida realize a multiplicação dessas matrizes armazenando o resultado em uma terceira matriz C [X, Z]. Ao final escrever a matriz C resultante.

- 4) Faça um programa que permita a uma secretária gerenciar os cadastros de pacientes de um consultório médico. As fichas contém: nome do paciente, sexo, idade, endereço, telefone, e uma lista das cinco últimas consultas. Para cada uma das consultas, são vinculados os diagnósticos finais de cada uma. Exemplo:

Nome	João da Silva		
Sexo	M	Idade	42
Endereço	Ruas das acácias, 400, Maringá - PR		
Telefone	(44) 2563 - 4141		
Consultas	15/12/2005	Resfriado	
	21/06/2006	Hepatite	
	22/09/2006	Anemia	
	14/02/2007	Virose	
	21/03/2007	Dengue	

O programa deve permitir que a secretária:

- 1) inclua um novo paciente;
- 2) exclua um paciente (mediante confirmação dos dados e não deixando espaços vazios no vetor de registros);
- 3) consulte a ficha de um paciente (fornecendo o nome);
- 4) liste todos os pacientes que já tiveram o diagnóstico de uma doença (fornecendo a doença);
- 5) saiba qual é a média de idade dos pacientes que já passaram por determinada doença (fornecendo a doença);
- 6) atualize a ficha de um paciente cadastrando uma nova consulta (fornecendo o dia e o diagnóstico e eliminando a consulta mais antiga caso a ficha estiver cheia);
- 7) atualize qualquer outro dado do cadastro que não seja uma nova consulta (mediante confirmação da atualização).

O programa deve guardar as fichas em um array de registros (as consultas podem ser implementadas em uma matriz ou array de registros). O programa deve mostrar um menu de opções para a secretária que irá escolher uma operação. O programa executará a operação, perguntando pelos dados necessários e mostrará novamente as opções do menu até que a secretária escolha a opção 'sair'.

Dica: implemente (no mínimo) um procedimento/função para cada uma das opções do menu.