3

Faculdade de Computação

Programação Procedimental 2ª Lista de Exercícios Preparatórios

Data de Entrega: 17/05/2016

Prof. Cláudio C. Rodrigues

Problemas:

P1. Escreva em linguagem C uma função denominada *telhado* que receba como parâmetros uma matriz **A**_{nxn} e um valor inteiro e ímpar (**n**) que define a dimensão da matriz nxn. A função telhado deve preencher a matriz recebida **A**_{nxn} como ilustrado nos exemplos abaixo. Elabore a função main() para testar a função telhado. O programa deve ler do dispositivo padrão de entrada (teclado) a dimensão **n** da matriz **A** e mostrar o resultado na tela de saída.

Exemplos	
Para $n = 5$	Para n = 9
0 1 2 1 0	0 1 2 3 4 3 2 1 0
1 0 1 0 1	1 0 1 2 3 2 1 0 1
2 1 0 1 2	2 1 0 1 2 1 0 1 2
1 0 1 0 1	3 2 1 0 1 0 1 2 3
0 1 2 1 0	4 3 2 1 0 1 2 3 4
	3 2 1 0 1 0 1 2 3
	2 1 0 1 2 1 0 1 2
	1 0 1 2 3 2 1 0 1
	0 1 2 3 4 3 2 1 0

O algoritmo deve estar contido no arquivo "telhado.c".

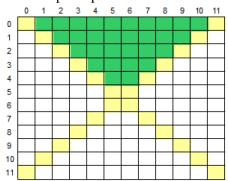
P2. Escreva uma função em linguagem C denominada *diag_superior* que receba como parâmetros uma matriz A_{nxn} de elementos reais de dupla precisão e um valor inteiro (\mathbf{n}) que define a dimensão da matriz nxn. Em seguida, calcule e mostre a média considerando somente aqueles elementos que estão na área superior da matriz, conforme ilustrado abaixo (área verde).

Entrada

Ler do dispositivo padrão de entrada a dimensão n e os valores de ponto flutuante de dupla precisão que compõem a matriz A_{nxn} .

Saída

Imprima média calculada, com 3 casas após o ponto decimal.



O algoritmo deve estar contido no arquivo "matrizverde.c".

P3. Dado um conjunto de **n** pontos de coordenadas (x_i,y_i) , o método de regressão linear permite determinar os coeficientes da reta $\mathbf{Y} = \mathbf{A}.\mathbf{X}+\mathbf{B}$ que se aproxima melhor do conjunto de pontos fornecidos; os coeficientes são determinados por

$$A = \frac{nS_{xy} - S_x S_y}{nS_{xx} - (S_x)^2}, \quad B = \frac{S_y - AS_x}{n}$$

Onde
$$S_x = \sum_{i=1}^n x_i$$
 e $S_{xy} = \sum_{i=1}^n x_i \cdot y_i$

Escreva em linguagem C uma função denominada *regressao_linear* que receba uma coleção de pontos (vetor de pontos) e a quantidade de pontos e retorna os coeficientes *A* e *B*. Construa a função *main()* para testar a função *regressao_linear* e apresentar os resultados na tela de saída.

O algoritmo deve estar contido no arquivo "regressao.c".

P4. A tabela abaixo apresenta os valores nutricionais de alguns alimentos comuns, considerando uma porção de 100g.

Alimento	Calorias	Glicídios	Lipídios	Proteínas
Pão	239,0	49,0	1,2	8,0
Arroz	354,0	77,0	1,7	7,6
Banana	90,0	20,0	0,5	0,4
Maça	52,0	12,0	0,3	0,3
Couve-flor	30,0	4,9	0,2	2,4
Tomate	22,0	4,0	0,3	1,0

Pretende-se calcular uma medida estatística de *correlação* entre duas sequencias de dados. Sejam $X=\{x_1,\,x_2,\,\ldots,\,x_n\}$ e $Y=\{y_1,\,y_2,\,\ldots,\,y_n\}$ as duas sequencias; supomos ainda que já calculamos as suas médias aritméticas $\overline{\boldsymbol{X}}$ e $\overline{\boldsymbol{Y}}$

$$\bar{X} = \frac{1}{n} \sum_{i=1}^{n} x_i \quad \bar{Y} = \frac{1}{n} \sum_{i=1}^{n} y_i$$

O fator de correlação $\rho(X,Y)$ é dado por

$$\rho(X,Y) = \frac{S_{XY}}{\sqrt{S_{XX}S_{YY}}}$$

Onde

$$S_{XY} = \sum_{\substack{i=1\\n}}^{n} (x_i - \bar{X})(y_i - \bar{Y})$$

$$S_{XX} = \sum_{\substack{i=1\\n}}^{n} (x_i - \bar{X})^2$$

$$S_{YY} = \sum_{\substack{i=1\\n}}^{n} (y_i - \bar{Y})^2$$

O *fator de correlação* determina valores no intervalo -1 e 1: onde o valor 1 significa total correlação; 0 nenhuma correlação e -1 total anticorrelação.

Escreva uma função em linguagem C que receba como parâmetros uma array A_{nxm} (tabela de valores nutricionais de uma coleção de alimentos) e a identidade de dois nutrientes a relacionar. A função deve calcular e retornar os <u>fatores de correlação</u>, por exemplo: $\rho(calorias, glicidios)$, $\rho(calorias, lipidios)$ e $\rho(calorias, proteinas)$.

O algoritmo deve estar contido no arquivo "correlacao.c".

P5. Escreva em linguagem C um programa chamado *extenso.c* que faça uso dos parâmetros *argv* e *argc*. O programa deverá receber da linha de comando um valor inteiro, e escrever no dispositivo padrão de saída (tela de display) o valor escrito por extenso na língua portuguesa.

Veja o exemplo, supondo que o executável se chame **extenso**: extenso 1234

O programa deverá imprimir:

Um mil duzentos e trinta e quatro

- **P6.** Escreva em linguagem C uma **função recursiva** que imprima a inversa de um string recebido como parâmetro.
- **P7.** Escreva em linguagem C uma **função recursiva** que conte o número de dígitos decimais que formam um número inteiro recebido como parâmetro.
- P8. Cadê o n-ésimo número? Seja s uma cadeia de caracteres, tal que cada caractere é um símbolo da tabela ASCII. Seja p > 0, o número que corresponde à posição ordinal de um número n ∈ ℵ em s. Construa em linguagem C uma função denominada (p_num) que retorna o p-ésimo n contido em s. Essa função deve receber p e s como parâmetros, e retornar n, se este número existir em s. Caso contrário, essa função deve retornar −1.

Observe o exemplo seguinte. Considere que s contém a cadeia de caracteres abaixo:



Considere também que a função se chame p_num , e as seguintes chamadas à mesma:

- 1. i = p-num(1,s);
- 2. j = p-num(2,s);
- 3. k = p-num(3,s).

Assim sendo, no caso 1, **i** deverá ser igual a 4, pois 4 é o primeiro número (p = 1) em s. No caso 2, **j** deverá ser igual a 1958, pois este é o segundo número (p = 2) em s. No caso 3, **k** deverá ser igual a -1, pois não existe um terceiro número (p = 3) em s.

Construa o programa principal. Declare as variáveis necessárias, colocando os comentários apropriados. Realize a chamada do procedimento cujo nome é p_num . Apresente os resultados obtidos na tela.

P9. Escreva em linguagem C um programa que declare uma matriz Mnxn de elementos inteiros, inicialmente preenchidos com o valor 0. A dimensão N deve ser definida pelo usuário. Elabore um algoritmo que leia do dispositivo padrão de entrada (teclado) os índices de linha e coluna de uma célula da matriz e incremente os valores dos elementos das posições ao redor desta célula de uma unidade (somar 1) se o valor do elemento for menor do que 9 (nove) e mude o valor do elemento da posição informada para 9 (nove).

Exemplo: linha 2 e coluna 3

00000	00000	Ex.: Observe ao lado o	000019
00000	001110	resultado do contorno	000011
00000	001910	para as posições (0,5)	0 1 1 1 0 0
00000	0 0 1 1 1 0	e (3,2)	019100

- **P10.** *Campo minado* é um jogo onde o computador sorteia **M** minas em um tabuleiro **NxP** e um jogador deve adivinhar onde as minas estão colocadas. Para isto um jogado realiza uma série de jogadas onde:
 - Jogador escolhe um quadrado do tabuleiro não descoberto do campo minado.
 - Jogador escolhe se revela o conteúdo da área ou declara como quadrado minado. Quando o jogador revela um quadrado vazio é exibido a quantidade de minas adjacentes a este quadrado que pode ser zero ou 8.

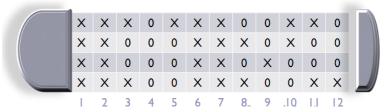
O jogo acaba quando o jogador revela uma mina (derrota) ou declara corretamente como "quadrado minado" todos os quadrados onde existem minas sem nenhum falso "quadrado minado".

A tarefa é implementar o jogo através das seguintes funções:

- uma função que imprime a situação atual do jogo. Áreas não reveladas devem ser representadas pelo caractere asterisco, áreas marcadas como minadas devem ser representadas com M e áreas reveladas sem minas devem ser representadas como o número de minas adjacentes se este valor for diferente de zero, caso este valor seja zero a região deve ser marcada como espaço em branco.
- uma função "Revelar" que retorna a quantidade de minas adjacentes ao quadrado escolhido ou −1 se existe uma mina no local
- Uma função "Marcar" que dada a posição no local verifica se existe uma mina no local e se tiver registrar como marcação verdadeira, senão como falsa.
- Dica: Use uma matriz para representar o tabuleiro e outra para os quadrados marcados e revelados.

P11. Uma empresa de transporte rodoviário possui ônibus com 48 lugares, distribuídos em 12 fileiras com quatro poltronas por fileira (duas janelas e dois corredores). Elabore um programa que controle as poltronas ocupadas no corredor e nas janelas. Considere que '0' representa poltrona livre e 'X' representa poltrona ocupada.

O programa deve controlar a venda de passagens da seguinte maneira: o programa mostra a situação da ocupação do ônibus e o usuário informa a poltrona escolhida.



P12. Exemplo de uso do método da Bisseção:

Bisseção: um método numérico para encontrar os zeros de uma função

O problema de calcular as raízes de uma equação sempre foi objeto de estudo da matemática ao longo dos séculos. No século XVII, o matemático norueguês, Niels Abel contribuiu com vários resultados notáveis e importantes para o desenvolvimento da matemática, provou que não existe uma fórmula geral para o cálculo das raízes exatas de uma equação polinomial de grau maior ou igual a 5. Nesses casos, e mesmo em casos mais simples, muitas vezes é necessário recorrer a métodos numéricos para calcular aproximações para as raízes reais de uma dada equação.

Existem vários métodos recursivos ou iterativos (do latim *iterare* = repetir, fazer de novo) para calcular aproximações numéricas para as raízes reais de uma equação.

Esses métodos consistem em, partindo de uma estimativa inicial, repetir o mesmo procedimento várias vezes, usando-se a cada vez como estimativa o resultado obtido na vez anterior, isto é na última iteração feita, até se alcançar a precisão desejada. Abaixo descrevemos o método conhecido como da *Bisseção*.

Método da Bisseção

Este método consiste em encontrar por inspeção dois pontos x_0 e x_1 tais que $f(x_0)$ e $f(x_1)$ tenham sinais contrários. Se $f(x_0)=0$ ou $f(x_1)=0$ você encontrou a raiz procurada. Caso contrário, existe pelo menos uma raiz de f(x)=0, entre x_0 e x_1 .

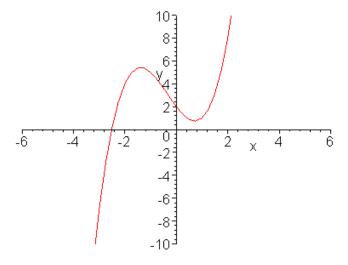
A figura ao lado ilustra este raciocínio. É fácil ver que f(-4) < 0 e que f(-2) > 0. Assim, existe um zero da função neste intervalo.

Considere agora o ponto $x_2=(x_0+x_1)/2$. Somente três casos podem acontecer: se $f(x_2)=0$, a raiz procurada é igual a x_2 , caso contrário, ou $f(x_2)$ e $f(x_1)$ têm sinais contrários e a raiz está entre x_2 e x_1 ou $f(x_2)$ e $f(x_0)$ têm sinais contrários e a raiz está entre x_2 e x_0 . Em qualquer dos casos a raiz pertence a um intervalo cujo comprimento é a metade do comprimento do intervalo anterior.

Repetindo-se o mesmo procedimento, encontra-se uma aproximação para a raiz da equação com a precisão desejada.

A cena abaixo permite que você entenda melhor este método. Nela, você pode modificar os parâmetros a_5 , a_4 , a_3 , a_2 , a_1 e a_0 que definem a função polinomial $y = a_5 x^5 + a_4 x^4 + a_3 x^3 + a_2 x^2 + a_1 x + a_0$. Variando estes parâmetros, você pode utilizar a cena para achar os zeros de qualquer função polinomial até o quinto grau. Use a representação gráfica da função, para determinar o intervalo onde existe um zero e escolher os valores iniciais para x_0 e x_1 de tal forma que $f(x_0) < 0$

e $f(x_1) > 0$. Entre com esses valores nos campos correspondentes. Automaticamente, o programa calculará os valores de x_2 , ponto médio do intervalo $[x_0, x_1]$, de $f(x_0)$, $f(x_1)$ e $f(x_2)$. Dependendo do sinal de f nesses três pontos, escolha o novo intervalo, $[x_0, x_2]$ ou $[x_2, x_1]$, onde estará localizado o zero procurado. Entre com o novo valor, substituindo o anterior para x_0 ou x_1 , conforme o caso, e comece tudo outra vez. O parâmetro i é um contador. Ele conta o número de interações realizadas. Você deve alterar o seu valor, a cada vez que entrar com um novo valor para um dos extremos do intervalo.



- Elabore uma função denominada get_raizes que percorra um intervalo do eixo \mathbf{x} [xi, xj] e procura as raízes da função $f(\mathbf{x})$ no intervalo. A função get_raizes deve dividir o intervalo [xi,xj] em subintervalos e avaliar em cada subintervalo a possibilidade da existência de uma raiz f(x)=0, conforme o método da bisseção.
- Elabore a função *bissecao* que recebe como parâmetros os limites de um subintervalo e devolve como retorno o valor da raiz (zero da função) obtida iterativamente, cuja método de aproximação foi apresentado acima. Dizemos que um determinado valor \mathbf{x}_i é raiz da função, se e somente se $f(x_i) = \mathbf{0}$.
- Elabore uma função denominada fx que implemente a equação polinomial desejada. A função fx deve receber como parâmetro o valor de x e retornar o valor f(x).