

Lista 03

Ponteiros

- **Atenção:**

1. **Identificadores de variáveis:** escolha nomes apropriados;
2. **Documentação:** inclua cabeçalho, comentários e indentação no programa.
3. **Arquivo-base:** você deve usar o arquivo-fonte incompleto fornecido junto com a lista. É necessário completar as operações nos lugares indicados e você não deve realizar nenhuma alteração nas partes fornecidas. Inclusive, se houverem comandos de entrada (`scanf`) e saída (`printf`) definidos, estes não poderão ser alterados.

- **Exercícios:**

1. Implemente uma calculadora geométrica. As opções que a calculadora deve ter e os detalhes mais relevantes são fornecidos abaixo:
 1. **Círculo:** a função recebe o raio r e deve retornar os valores da área (πr^2) e circunferência ($2\pi r$), passados como parâmetros.
 2. **Esfera:** a função recebe o raio r e deve retornar os valores de volume ($\frac{4}{3}\pi r^3$) e área de superfície ($4\pi r^2$), passados como parâmetros.
 3. **Retângulo:** a função recebe o comprimento c e a largura l e deve retornar os valores de área (lc) e perímetro ($2l + 2c$), passados como parâmetros.
 4. **Cubo:** a função recebe a aresta a e deve retornar os valores de volume (a^3), área de superfície ($6a^2$) e diagonal ($a\sqrt{3}$), passados como parâmetros.
 5. **Cone:** a função recebe o raio r e a altura h e deve retornar os valores de volume ($\pi r^2 \frac{h}{3}$) e área de superfície ($\pi r(r + \sqrt{h^2 + r^2})$).
 6. **Pirâmide:** a função recebe o comprimento c , a largura l e a altura h e deve retornar os valores da superfície lateral ($c\sqrt{(\frac{l}{2})^2 + h^2} + l\sqrt{(\frac{c}{2})^2 + h^2}$) e do volume ($\frac{hlc}{3}$), passados como parâmetros.
 0. **Sair:** finaliza o programa.

Detalhe

- (a) As entradas são dadas por números reais positivos.

Complete o arquivo L03EX01.c

Exemplos de E/S (os comentários entre parênteses não deverão ser exibidos):

Entrada	Saída
1 2	12.57 12.57
2 3	113.09 113.09
3 5 6	30.00 22.00
4 2	8.00 24.00 3.46
5 4 5	83.77 130.73
6 2 3 5	25.74 10.00
0	

2. Faça um programa que troca o valor de três variáveis baseado em uma cadeia de caracteres. O programa deve receber três inteiros (os valores que serão trocados), um número natural $n \leq 20$ indicando o tamanho de uma cadeia de caracteres e a própria cadeia de caracteres. Após realizar todas as trocas, o programa ainda deve informar se os valores armazenados nas variáveis estão ordenados ou não. Para ilustrar como funciona o procedimento de troca baseada na cadeia de caracteres, veja o seguinte exemplo:

Passo	int A	int B	int C	Cadeia de Caracteres
Inicial	1	15	8	ABCA
1	15	1	8	A BCA
2	15	8	1	AB C A
3	1	8	15	ABCA B

Dentro do código-fonte base há duas variáveis “ordenado” e “desordenado.” Estas variáveis são passadas por ponteiro para a função que realizará as trocas. Ao final das trocas, a função deverá retornar um ponteiro. Se os valores das variáveis estiverem ordenados, ou seja $A \leq B \leq C$, o ponteiro deverá apontar para a variável “ordenado” e o programa imprimirá 1. Caso contrário, o ponteiro deverá apontar para a variável “desordenado” e o programa imprimirá -1.

Complete o arquivo L03EX02.c

Exemplos de E/S (os comentários entre parênteses não deverão ser exibidos):

Entrada	Saída
1 8 3 2 BC	1 3 8 (1)

3. Uma imagem é uma matriz de cores, pode ser representada por um número inteiro. Cada posição da matriz corresponde a um pixel da imagem, sendo que o valor de cada pixel representa a sua cor. Para imagens em escalas de cinza, estes valores variam de 0 (preto) a 255 (branco).
- Faça um programa que manipule uma imagem (matriz) quadrada em escala de cinza. A manipulação sobre a matriz deverá ser feita utilizando um ponteiro que apontará para a primeira posição da matriz (já está implementado).
- Seu programa deverá primeiramente receber a ordem da matriz (representando a quantidade de pixels na largura e no comprimento da imagem) e seus respectivos valores (cores). Em seguida, a matriz deverá ser exibida. Então, o usuário poderá escolher as seguintes opções:
- (0) **Sair:** encerra a execução do programa
 - (1) **Clarear:** recebe uma matriz A (de mesma dimensão que a matriz inicial M) e calcula $M = M + A$
 - (2) **Escurecer:** recebe uma matriz A (de mesma dimensão que a matriz inicial M) e calcula $M = M - A$
 - (3) **Diagonal:** informa se a matriz M é uma matriz diagonal
 - (4) **Transpor:** substitui a matriz M por sua transposta (rotaciona a imagem)
 - (5) **Binarizar:** aplica a operação de binarização, recebendo um número como parâmetro e fazendo com que todos os campos da matriz maiores ou iguais a este número recebam 255, enquanto os outros campos recebem 0 (a imagem fica em preto e branco)

- (6) **Maior e Menor:** exibe o maior e menor valor da matriz (cor mais clara e mais escura da imagem). Repare que esta função já está implementada: o maior e menor valor devem ser apontados pelos ponteiros globais `*maior` e `*menor`

Complete o arquivo L03EX03.c

Detalhes

- (a) As exibições já estão pré-definidas no código-fonte base fornecido.
- (b) Preste atenção e implemente as funções que estão sem implementação no código-fonte base.
- (c) Como os valores da matriz variam de 0 a 255, qualquer número negativo deve ir para zero e qualquer número maior que 255 deve ir para 255.

Exemplos de E/S (os comentários entre parênteses não deverão ser exibidos):

Entrada	Saída
2	(Dimensões)
1 2	(Matriz)
3 4	
	001 002 003 004
1	(Opção: Clarear)
1 0	
0 254	
	002 002 003 255
2	(Opção: Escurecer)
3 0	
0 0	
	000 002 003 255
3	(Opção: Diagonal)
	NAO
4	(Opção: Transpor)
	000 003 002 255
5 3	(Opção: Binarizar com parâmetro = 3)
	000 255 000 255
6	(Opção: Maior e Menor)
	Maior -i 255 Menor -i 0
0	

Casos de teste

- (a) Leitura da imagem, Clarear e Escurecer
- (b) Caso anterior + Diagonal
- (c) Caso anterior + Transpor
- (d) Caso anterior + Binarizar
- (e) Todas as opções