

Lista de Exercícios N° 1

Esta lista pode ser desenvolvida em grupos de até dois alunos, seguindo as especificações contidas no arquivo 00_ProcListas2.pdf disponibilizado pelo professor na área de arquivos da equipe da disciplina na plataforma Teams. A entrega ao professor deverá ocorrer até o dia 22 de abril de 2025.

- 1) Escreva um programa que recebe duas palavras de até 12 caracteres cada digitadas pelo usuário. Caso as duas palavras sejam exatamente iguais, possuam pelo menos 6 caracteres, pelo menos uma letra em minúscula, pelo menos uma letra em maiúsculas e pelo menos um dígito de 0 a 9, o programa deverá emitir a mensagem “Nova senha ok” e, em caso contrário, imprimir a mensagem “Nova senha invalida”.
- 2) Criar uma *struct* para representar o tipo ponto, por meio da seguinte instrução:

```
struct ponto { float x; float y; } ;
```

Em seguida, o programa deverá receber vários pares de pontos e, para cada par, determinar a distância entre eles, conforme a fórmula

$$\text{Distância} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Encerrar o processamento quando for recebido um conjunto de entradas com quatro números zero.

Exemplos

Entrada:

1.0	5.0	3.0	4.2
1.0	5.0	3.0	1.5
1.0	5.0	6.4	2.45
3.0	1.5	7.0	2.2
0.0	0.0	0.0	0.0

Saída esperada:

2.154066
4.031129
5.971809
4.060788

- 3) Faça um programa para verificar se um ponto está contido dentro de um retângulo ou não. Receber inicialmente uma quantidade inteira N que indicará o número de casos de teste. Para cada caso de teste, receber reais x_0, y_0, x_1, y_1, x, y onde (x_0, y_0) corresponde ao vértice inferior esquerdo do retângulo, (x_1, y_1) ao vértice superior direito e (x, y) são as coordenadas do ponto a ser verificado. Receber os valores e imprimir a resposta (“Contido” ou “Nao-contido”) na rotina `main()` e usar uma *struct* para armazenar as duas coordenadas do ponto.

Exemplos

Entrada:

3
1.0 1.0 5.0 4.0 3.0 2.0
1.1 1.3 5.4 4.5 6.0 2.1
1.2 0.0 3.4 6.5 3.2 5.7

Saída esperada:

Contido
Nao-contido
Contido

Lista de Exercícios N° 1

- 4) Escreva um programa em linguagem C que recebe os dados fornecidos pelo usuário e os armazena nas matrizes PecaMes e GranaPeca indicadas abaixo. A primeira indica a produção das peças P1, P2 e P3 nos meses de Janeiro a Março e a segunda indica o custo de cada tipo de peça bem como a respectiva margem de lucro. O programa deve então gerar e exibir na tela a matriz GranaMes, que indica os valores de custo e lucro verificados de Janeiro a Março.

<u>PecaMes</u>				<u>GranaPeca</u>		
	P1	P2	P3		CUSTO	LUCRO
JAN	1	2	3	P1	1	0
FEV	4	5	6	P2	2	1
MAR	7	8	9	P3	4	2

<u>GranaMes</u>		
	CUSTO	LUCRO
JAN	17	8
FEV
MAR

- 5) Faça um programa que recebe 10 valores inteiros e os armazena em um vetor. Em seguida, o programa deve verificar se o vetor já está ordenado de forma ascendente: caso esteja, imprimir a mensagem “Vetor já ordenado” e, em caso contrário, ordenar o vetor através do método bolha, imprimir seu conteúdo ordenado e também a quantidade de trocas que foram necessárias para fazer a ordenação.

Exemplo 1

Entrada:

23 13 35 29 18 45 1 29 47 20

Saída esperada:

1 13 18 20 23 29 29 35 45 47

19 trocas foram necessarias

Exemplo 2

Entrada:

5 12 18 26 31 31 35 39 44 45

Saída esperada:

Vetor ja ordenado