

## Postar dentro do algoritmo o nome da dupla

01) Realize as seguintes atividades:

- Crie um algoritmo para inserir “n” códigos de produtos, “n” nomes de produtos e “n” valores de produtos.
- Crie uma função para exibir os produtos na tela, como no exemplo abaixo:

Código	Produto	Valor	Estoque
99	Arroz	12,50	10
45	Feijão	7,9	9
41	Cerveja	4,9	5

- Crie uma função onde o usuário digite um código e verifique se este código de produto está cadastrado, se houver exiba o código o nome, valor do produto e a quantidade em estoque;
- Crie uma função onde o usuário digite o código de produto, se encontrar, informe a quantidade de compra, caso contrário, avise para o usuário que a quantidade não existe em estoque,
- Se tiver no estoque, armazene o código e a quantidade em uma matriz (ou vetor) de venda;
- Pergunte se ele deseja inserir mais produtos;
- Caso o usuário não desejar mais realizar compras, exiba o nome de todos os produtos comprados, suas respectivas quantidades e o total da compra.
- Por fim, exiba novamente a situação dos produtos na tela (código, nome, valor e estoque).

Exemplo:

Produto	Qtd	SubTotal
Arroz	2	25,00
Feijão	1	7,9
Cerveja	10	49,00

## Lista de Exercício nº 4 – Sub-rotinas e Matrizes

Total da compra: 81,9 R\$

02)DESAFIO:

## Lista de Exercício nº 4 – Sub-rotinas e Matrizes

### **Recriando o PACMAN**

*Arquivo fonte:* pacman.{c | cpp | java | py | cs}

Matoso e Demétrio estão revivendo os tempos de Atari e resolveram criar um novo jogo com base no PACMAN. Para quem não se lembra, o PACMAN era um personagem que andava por um labirinto comendo diversas “quadrados” e nos cantos da tela haviam algumas vitaminas para deixar a velocidade do personagem mais rápida, porém havia alguns fantasmas coloridos que ao encostar no personagem o agarravam e você perdia o jogo.

Pois bem, um dos princípios básicos de qualquer jogo é a detecção de colisão. O objetivo, basicamente, é verificar se dois objetos quaisquer colidiram (o fantasma com o PACMAN).

Para facilitar as coisas, muitas vezes os objetos são aproximados por figuras geométricas simples (esferas, paralelepípedos, triângulos etc). Neste problema, os objetos são aproximados por retângulos num plano 2D.

#### **Tarefa**

Escreva um programa que, dados dois retângulos, determine se eles se interceptam ou não.

#### **Entrada**

A entrada contém um único conjunto de testes, que deve ser lido do dispositivo de entrada padrão (normalmente o teclado). Cada caso de teste contém duas linhas. Cada linha contém quatro inteiros ( $x_0$ ,  $y_0$ ,  $x_1$ ,  $y_1$ , sendo  $0 \leq x_0 < x_1 \leq 1.000.000$  e  $0 \leq y_0 < y_1 \leq 1.000.000$ ) separados por um espaço em branco representando um retângulo. Os lados do retângulo são sempre paralelos aos eixos  $x$  e  $y$ .

#### **Saída**

Seu programa deve imprimir, na saída padrão, uma única linha para cada caso de teste, contendo o número 0 (zero) caso não haja interseção ou o número 1 (um) caso haja.

Exemplo:

Entrada	Saída
0 0 1 1 0 0 1 1	1

Entrada	Saída
0 0 2 2	1

## Lista de Exercício nº 4 – Sub-rotinas e Matrizes

1 1 3 3	
---------	--

Entrada	Saída
0 0 1 1 2 2 3 3	0