

[C04] Exercícios de Revisão

A) Batalha de gladiadores

A Sony, a principal empresa de videogames do mundo, está recrutando desenvolvedores de jogos para integrar seu seleto time de desenvolvimento. Para selecionar os melhores talentos, eles propuseram um desafio de programação, no qual os participantes têm a oportunidade de começar suas carreiras no emocionante mundo dos games.

O desafio consiste em desenvolver um sistema de batalha entre gladiadores, que opera da seguinte maneira:

1. O jogador deve cadastrar 5 gladiadores, fornecendo os seguintes atributos para cada um: Nome (String), Força (int), Habilidade (int) e Idade(int). Tanto a força quanto a habilidade variam de 0 a 10.
2. Após cadastrar os 5 gladiadores, o jogador pode escolher quais gladiadores irão se enfrentar.
3. As batalhas ocorrem da seguinte forma:
 - a. Cada gladiador inicia com 100 pontos de vida.
 - b. O primeiro gladiador selecionado ataca primeiro.
 - c. O dano causado durante um ataque é calculado como "DANO = força + (2^habilidade)".
 - d. Após o ataque do primeiro gladiador, é a vez do próximo gladiador, e assim por diante, até que um dos dois seja derrotado (quando sua vida atinge 0).
4. Para sair do jogo, o jogador deve digitar '-1'.
5. Ao final do código, forneça comentários indicando a complexidade assintótica do seu código e a complexidade, no modelo RAM, das funções do seu código (Coloque como comentários no próprio código).

Dicas:

1. *Utilize uma struct para armazenar os gladiadores.*
2. *Separe a solução do exercício em funções para facilitar a resolução e deixar seu código mais organizado. Por exemplo:*

```
bool combate(gladiador g1, gladiador g2){  
    //Código que retornará 0 caso g1 ganhe ou 1 caso g2 ganhe  
}
```

Exemplo de Entrada:	Exemplo de Saída:
----------------------------	--------------------------

Marcelo 10 10 21 Goku 8 7 38 Gon 5 4 12 Gojo 7 8 28 Luffy 6 7 17 0 1 3 4	Marcelo ganhou a batalha contra Goku. Gojo ganhou a batalha contra Luffy.
---	--

B) Colecionador de insetos

Anderson finalmente conseguiu um espaço em seu apartamento para expor sua coleção de insetos, e gostaria de fazê-lo de forma organizada. Cada inseto está guardado dentro de uma caixa de vidro, e rotulado com o nome científico da espécie e o tamanho em centímetros de cada um. Para otimizar seu tempo, Anderson pediu a você para que faça um programa que, dados a quantidade de insetos que ele possui e a informação de cada rótulo, ordene seus insetos por tamanho, e retorne apenas o nome da espécie na ordem desejada.

Exemplo de Entrada:	Exemplo de Saída:
5	Apis mellifera

Bombus terrestris 2 Apis mellifera 1.5 Mantis religiosa 7.5 Danaus plexippus 3.5 Anoplophora glabripennis 3	Bombus terrestris Anoplophora glabripennis Danaus plexippus Mantis religiosa
Exemplo de Entrada: 8 Pieris brassicae 2.2 Lucanus cervus 7 Papilio machaon 3.2 Tenebrio molitor 1.5 Forficula auricularia 1.5 Nymphalis antiopa 3 Zophobas morio 2.5 Periplaneta americana 4	Exemplo de Saída: Tenebrio molitor Forficula auricularia Pieris brassicae Zophobas morio Nymphalis antiopa Papilio machaon Periplaneta americana Lucanus cervus
Exemplo de Entrada: 7 Coccinella septempunctata 0.8 Acherontia atropos 5 Oryctes nasicornis 4 Vanessa atalanta 2.8 Gryllus campestris 2.5 Aglais io 2.3 Melolontha melolontha 3	Exemplo de Saída: Coccinella septempunctata Aglais io Gryllus campestris Vanessa atalanta Melolontha melolontha Oryctes nasicornis Acherontia atropos

--	--

C) Pastelaria do Dudu (Sequencial)

Eduardo estava cuidando do estoque da pastelaria da família e teve a ideia de chamar um aluno do Inatel para ajudar a registrar os itens para verificar com mais facilidade se possui ou não determinado produto. Ajude Eduardo a criar um sistema que armazene nome e quantidade de cada produto em um vetor, e imprima uma mensagem informando se existe ou não em estoque.

Entrada

A primeira linha consiste num valor inteiro N correspondente à quantidade de produtos cadastrados. E as demais N linhas consistem em uma string correspondente ao nome do produto e um valor inteiro correspondente à quantidade (unidade(s) ou pacote(s)) em estoque. Em seguida deve ser colocado o nome do produto que deseja buscar.

Saída

A saída consiste em uma mensagem dizendo se tem em estoque ou não. O produto deve ser impresso na mensagem.

Exemplo de entrada: 4 Farinha 2 Ovo 12 Oleo 1 Sal 1 Ovo	Exemplo de saída: Ovo em estoque!
Exemplo de entrada: 3 Frango 5 Calabresa 4 Queijo 9 Banana	Exemplo de saída: Nao existe Banana em estoque!

D) Pastelaria do Dudu (Binária)

Eduardo, após refletir muito, achou que poderia melhorar o sistema do estoque adicionando um código em ordem crescente em cada produto cadastrado, dessa forma, você pode usar um algoritmo de Busca Binária para facilitar a busca quando o estoque estiver cheio. Ajude Eduardo a criar esse novo sistema que armazene o código do produto, nome e quantidade em um vetor, e imprima uma mensagem informando se existe ou não.

Entrada

A primeira linha consiste num valor inteiro N correspondente à quantidade de produtos cadastrados. E as demais N linhas consistem em um valor inteiro correspondente ao código do produto, uma string correspondente ao nome do produto e um valor inteiro correspondente à quantidade (unidade(s) ou pacote(s)) em estoque. Em seguida deve ser colocado o código do produto que deseja buscar.

Saída

A saída consiste em uma mensagem dizendo se tem em estoque ou não. Se houver, o produto deve ser impresso na mensagem.

Exemplo de entrada: 4 1 Farinha 2 2 Ovo 12 3 Oleo 1 4 Sal 1 2	Exemplo de saída: Ovo em estoque!
Exemplo de entrada: 3 1 Frango 5 2 Calabresa 4 3 Queijo 9 7	Exemplo de saída: Nao existem produtos com esse codigo em estoque!

E) Pilha e Hashing

Considere a inserção de elementos em uma pilha e quando um elemento for removido ele deve ser inserido em uma tabela hash de comprimento M usando endereçamento aberto. Mostre o resultado da inserção dessas chaves utilizando hash duplo com $h_1(k) = k \bmod m$ e $h_2(k) = 1 + (k \bmod (m - 1))$.

A entrada consiste em três linha onde a primeira é o valor de M, a segunda é a quantidade de elementos que serão inseridos e a terceira são os elementos que devem ser armazenados.

A saída deve mostrar os elemetos depois de serem inserido na tabela hash.

obs.: Nas posições que não forem armazenado nenhum elemento deve mostrar -1.

Exemplo de entrada	Exemplo de saída:
--------------------	-------------------

11 9 10 22 31 4 15 28 17 88 59	88 10 28 22 59 -1 17 -1 31 4 15
7 4 12 23 6 43	-1 43 23 -1 -1 12 6

F) Loja de Itens

Poles está organizando os itens disponíveis para venda em sua loja online. Ele quer criar um sistema de compra e venda de itens.

Para isso ele tem uma lista com diversos itens e suas informações (código, nome e preço). Ele precisa cadastrar essa lista em uma tabela hash de tamanho $M = 43$ para facilitar as buscas. A função hash usada para a inserção deve ser pelo método do módulo, considerando o código do item e o valor M , com tratamento de colisão por sondagem linear.

Em seguida ele quer que, dada uma lista de itens sejam inseridos em uma fila de vendas e uma lista de itens sejam inseridos em uma pilha de compras.

Por fim, ele vai entrar com dois números inteiros V e C , indicando respectivamente a quantidade de vendas e compras ele quer realizar. As vendas e compras devem ser processadas das respectivas listas. Ele quer saber o lucro total das operações. Sabe-se que o lucro L é o valor total VT das vendas realizadas subtraído do valor total VC das compras realizadas ($L = VT - VC$).

Entrada

A entrada é dividida em quatro partes. Na primeira parte é informado um número inteiro N ($0 < N \leq 43$) que é a quantidade de itens e em seguida são entradas N linhas, cada uma contendo as informações de cada item i , sendo elas um número inteiro indicando o código do item, uma cadeia de caracteres (sem espaço) indicando o nome do item e um número inteiro indicando o preço do item. Na segunda parte serão informados vários números inteiros vi , indicando os códigos dos itens a serem vendidos (a entrada termina com $vi = -1$). Esses itens devem ser inseridos em uma fila. Na terceira parte serão informados vários números inteiros ci , indicando os códigos dos itens a serem comprados (a entrada termina com $ci = -1$). Esses itens devem ser inseridos em uma pilha. Na última parte, são informado dois números inteiros V e C indicando respectivamente a quantidade de itens a serem processados da fila dos vendidos e a quantidade de itens a serem processados da pilha dos comprados.

Saída

Deve exibir um número inteiro com o lucro total obtido seguido dos caracteres " moeda(s)".

Exemplo de entrada: 4 3477 arco 20 5455 escudo 15 2125 espada 12 3474 flecha 10 3477 5455 -1 2125 3474 -1 1 0	Exemplo de saída: 20 moeda(s).
Exemplo de entrada: 4 3477 arco 20 5455 escudo 15 2125 espada 12 3474 flecha 10 3477 5455 -1 2125 3474 -1 2 1	Exemplo de saída: 25 moeda(s).