

Prova de Introdução à Programação

Prof Alexandre Mota - 21/08/2023

Questão. O CIn está criando um serviço robótico semelhante à figura abaixo



onde temos um robô no centro de uma estrutura que lembra o coliseu de Roma. Abstraindo esta figura, obtemos uma matriz dinâmica parecida com um tronco de um cone (de cabeça para baixo). Suponha que a matriz que irá representar esta estrutura tenha na base 20 elementos (ou colunas) e a cada nível de altura (nova linha), a matriz aumenta 2 elementos. Esta matriz terá 20 linhas de altura (e 58 colunas na parte maior dela).

Para representar tal matriz em C, precisaremos usar ponteiros duplos. Cada célula desta matriz irá conter uma quantidade igual de determinado medicamento. A localização do medicamento será dada por um arquivo texto (um arquivo de configuração), cuja linha terá a linha e a coluna do medicamento, seguida de seu nome, todos separados por vírgulas e a linha terminando com '\n'. Em um segundo arquivo, agora binário, seu programa em C encontrará uma lista de pedidos de medicamentos, onde cada pedido é um vetor dinâmico de nome de medicamento e quantidade. Visando facilitar a leitura do arquivo, ele foi construído de tal forma que tem-se duplas com o tamanho do vetor dinâmico de um pedido, seguido do pedido (vetor em si) até o final do arquivo, como na figura abaixo

2	Medic1 (1) Medic2 (3)	4	Medic5 (2) Medic9 (1) Medic3 (5) Medic7 (2)
---	-----------------------	---	---

Seu programa deve usar as seguintes funções para fazer este serviço funcionar:

- `localM *carregaConfig(int *tamC)`: lê linha, coluna e nome do medicamento do arquivo "config.txt" em vetor dinâmico do tipo `localM` (estrutura `int linha, coluna; char medic[50];`), retornando este vetor dinâmico de tamanho `tamC`;
- `pedido *carregaPedidos(int *tamP)`: lê arquivo binário de a figura acima, retornando vetor dinâmico do tipo `pedido` (estrutura `int qtd; outra estrutura do tipo TMed (int qtdM; char medic[50];)`) e o tamanho deste vetor em `tamP`;
- `coliseu **inicializa()`: retorna ponteiro duplo de 20 linhas por 20 a 58 colunas, onde cada célula terá um número inteiro igual a 20 (20 unidades do medicamento);
- `void processaPedido(pedido meuPedido, coliseu **estoque, localM *config, int tamC)`: adiciona em arquivo texto (pedidos.txt) este pedido no formato "separei N unidades do medicamento M", diminuindo a quantidade N na matriz `coliseu`, a partir da configuração `config`;
- `void processaPedidos(pedido *todosPedidos, int tamP, coliseu **estoque, localM *config, int tamC)`: adiciona em arquivo texto (pedidos.txt) a linha "++..." (com 70 caracteres '+') e chama a função `processaPedido`. Ao final, esta função libera todos os ponteiros;

E ter na main todos os ponteiros usados neste programa. Abra e feche os arquivos nos locais mais convenientes.

Boa sorte!!!