2019/02

Lista de Exercícios de Revisão de C (alocação dinâmica, estruturas de dados)

1) Uma matriz é classificada como diagonal, se todos os elementos que não estão na diagonal principal forem zero. Triangular inferior é a matriz com a seguinte propriedade: todos os elementos acima (não inclusive) da diagonal principal têm valor zero. Quando todos os elementos abaixo da mesma diagonal têm valor zero, a matriz é classificada como triangular superior.

Criar um programa que receba do usuário o tamanho da matriz e crie a mesma utilizando alocação dinâmica. Nessa alocação dinâmica, os dados (double) da matriz devem ficar em um espaço contíguo da memória. Após isso, o usuário irá entrar com os dados da matriz.

O seu programa deve também criar um array de ponteiros. Esse array conterá 3 ponteiros para funções que recebem uma matriz com o devido tamanho e retornam um inteiro. A primeira função retorna 0 se a matriz não for diagonal e 1 caso contrário. A segunda faz o mesmo para matriz triangular inferior e a terceira para matriz triangular superior. Um outro procedimento deve ser criado, que recebe a matriz com o tamanho e a função que dever ser aplicada. Esse procedimento aplica a função e escreve na tela: “a matriz é do tipo selecionado” se o resultado for 1 e “não é do tipo selecionado” se o resultado for 0. O programa principal, depois de ler os dados da matriz, deve receber do usuário qual o tipo de teste a ser feito (1-diagonal, 2-triangular inferior, 3-triangular superior) e deve chamar esse procedimento passando como parâmetro o endereço da função solicitada.

No final, o conteúdo da matriz deve ser impresso na tela utilizando para isso um ponteiro que anda em toda estrutura de dados contínua da matriz imprimindo seu conteúdo.

2) Fazer um programa que receba uma lista de nomes de usuário de um SO juntamente com seu identificador uid. Uma lista encadeada ordernada por uid deve ser criada para guardar essas informações. Após isso, uma lista de processos juntamente com o id do usuário que o criou é entrada pelo usuário. Dentro de cada nó da lista de nomes de usuário, deve ser armazenado o ponteiro de início de uma outra lista, a lista de processos do respectivo usuário. No final deve ser apresentado cada usuário, seu id e a lista de processos que estão executando com seu uid.

Exemplo de entrada:
4
user4 4
user1 1
user3 3
user2 2
6
progA 2
progB 1
progC 2
progD 2
progE 4
progF 2

Exemplo de saída:
user1 1
progB
user2 2
progA

progC
progD
progF
user3 3
user4 4
progE