

Universidade Federal do Maranhão
Departamento de Informática
Disciplina: Estrutura de Dados
Prof. Anselmo Paiva

Alexsandro da Silva Soriano

Primeira Avaliação

1. Faça um algoritmo para receber uma matriz $n \times m$ armazenada no vetor $v1$, e retornar 1 se a matriz é triangular inferior, -1 se a matriz é triangular superior e zero se é uma matriz diagonal.

`int DeterminaTipoMatriz (int *v1, int n, int m) ;`

OBS : não pode usar outros vetores ou matrizes

Matriz é triangular superior é aquela em que os elementos abaixo da diagonal principal são nulos: $\forall i > j, a_{ij} = 0$

Uma matriz é triangular inferior é aquela em que os elementos acima da diagonal principal são nulos: $\forall i < j, a_{ij} = 0$

Matriz é diagonal quando os elementos acima e abaixo da diagonal principal são nulos

2. Faça um algoritmo para receber uma fila armazenada em um vetor circular, e remover todos os elementos DA FILA que estão em posição ímpar (NA FILA). Por exemplo, remover o Primeiro, terceiro, quinto e sucessivamente.

`typedef struct _Fila_ {`

`void **elms; int maxElms; int beg, end ;`

`} Fila;`

`int RemoveElementosImpar (Fila *f);`

OBS: Não pode usar memória adicional; Fila não tem contador com número de elementos.

3. 3- faça um algoritmo que inverte os elementos de uma fila, de forma que o início se torne o fim e vice-versa. Para isso, use os TAD de Pilha e Fila abaixo (não precisa implementar as funções dessas TADs).

`Stack *stkCreate(int max); //Cria e aloca a pilha`

`int stkEmpty (Stack* s); //Retorna verdadeiro se a pilha está vazia`

`int stkPush(Stack* s, void *elm); //insere na pilha`

`void * stkPop(Stack* s); //Retira e retorna o elemento da pilha`

`Queue *qCreate(int max); //Cria e aloca a fila`

`int qEmpty (Queue* q); //Retorna verdadeiro se a fila está vazia`

`int qEnqueue(Queue * q, void *elm); //insere na fila`

`void * qDequeue(Queue * q); //Retira e retorna o elemento da fila`