

Defina as estruturas necessárias e faça um algoritmo para:

1. Escreva um algoritmo que recebe uma matriz  $AN \times N$  de inteiros, armazenada em um vetor  $v$ , e retorna em um vetor  $vcoluna$  todos os elementos da diagonal principal da matriz  $A$ .

int ExtraiColuna (int \*v, int \*vcoluna, int N, int C)

2. Faça um algoritmo que recebe uma árvore binária de pesquisa e conta quantos nós possuem um valor maior que um valor dado.

int abpContaMajores( TNode \*t, void \*key, int (\*cmp)(void \*, void \*))

3. Faça um algoritmo que recebe dois vetores de caracteres  $s1$  e  $s2$  com uma frase em que as palavras estão separadas por espaços em branco e, usando uma única pilha, verificar se as palavras que existem em  $s1$  estão em  $s2$  na mesma sequência mas em ordem inversa. Não pode usar memória auxiliar somente usar as funções do TAD (stkCreate, stkPop, stkPush, stkDestroy).

int VerificaString(Stack \*s1, char \*s2, int length);

Exemplo de strings que tornam verdadeira a afirmativa:

$S1 = \text{"O Flamengo é o melhor time do Brasil"}$

$S2 = \text{"O ognemalf é o rohlem emit od lisarB"}$

4. Escreva um algoritmo que recebe uma lista circular duplamente encadeada  $L$  e remove um elemento especificado pela chave  $Key$ , juntamente com seu vizinho (próximo).

Int RemoveOCaraEOsVizinhos( Dllist \*l, void \*key,  
int (\*cmp)( void\*, void\*))

