

Lista de exercícios sobre Strings e Matrizes - Lista11 - Para resolver estes exercícios você irá precisar ler o texto "Semana 11 - Matrizes - parte 1.pdf" e "Semana 11 - Matrizes - parte 2.pdf" disponível na Semana 11 do curso, e também pode consultar os slides dos encontros síncronos da semana 11 sobre Strings e Matrizes.

1. Faça um programa que leia uma frase, a qual pode haver ou não presença de letras maiúscula, e imprima esta frase usando apenas letras maiúsculas.
Dica: Utilize a tabela ASCII para projetar seu código. Faça sempre uma projeção de como será o seu código antes de começar programá-lo.

```
#include <stdio.h>
```

```
char maiuscula(char vetor[], int i){
```

```
    if (vetor[i] >= 'a' && vetor[i] <= 'z'){  
        vetor[i] = (vetor[i] - 'a') + 'A';  
    }  
    return vetor[i];  
}
```

```
int main(){  
    int i;  
    char letra, frase[61];  
    printf("Escreva uma frase: ");  
    scanf("%60[^\n]", frase);  
  
    for(i=0; frase[i]!='\0'; i++){  
        letra = maiuscula(frase, i);  
        printf("%c", letra);  
    }  
    return 0;  
}
```

2. Dados duas Strings S1 e S2, verifique quantas são as ocorrências da String S2 dentro de S1 e exiba esta informação na tela.

Exemplo: se S1 é a String “Raras araras em Araraquara” e S2 é a String “ara”, o seu programa deve responder 5, pois a String “ara” aparece uma vez em “Raras”, duas em “araras” e outras duas em “Araraquara” (e não três já que é feita a distinção entre maiúsculas e minúsculas).

Dica: Para resolver este problema, é útil você fazer uma função que verifica se uma string menor encaixa em uma string maior e registrar aonde isso ocorre. Faça sempre uma projeção de como será o seu código antes de começar programá-lo.

Tentaram me ajudar, mas mesmo assim o código não funcionou, se puder falar sobre o código antes da prova, ficarei grato!

```
#include <stdio.h>
```

```
#include <string.h>
```

```
int verificar(char s2[], char s1[], int ind){
```

```
    int i, j;
```

```
    i = ind;
```

```
    for(i = 0; s2[i] != 0; i++){
```

```
        if(s2[i] != s1[j]){
```

```
            return 0;
```

```
        }
```

```
        j++;
```

```
    }
```

```
    return 1;
```

```
}
```

```
int main(){
```

```
    char s1[201];
```

```
    char s2[201];
```

```
    int cont, i, tamanho1, tamanho2;
```

```
printf("Escreva a primeira string: ");  
scanf("%[^\n]", s1);
```

```
printf("Escreva a segunda string: ");  
scanf("%[^\n]", s2);
```

```
tamanho1 = strlen(s1);  
tamanho2 = strlen(s2);
```

```
for(i=cont=0; i < tamanho1 - tamanho2 + 1; i++){  
    cont = cont + verificar(s2, s1, i);  
}
```

```
printf("A segunda string aparece %d vezes na primeira string!", cont);
```

```
return 0;  
}
```

3. Dizemos que uma matriz quadrada inteira é um quadrado mágico se a soma dos elementos de cada linha, a soma dos elementos de cada coluna e a soma dos elementos das diagonais principais e secundaria são todas iguais.

Exemplo: A seguinte matriz 3×3 é um quadrado mágico.

$$\begin{pmatrix} 8 & 0 & 7 \\ 4 & 5 & 6 \\ 3 & 10 & 2 \end{pmatrix}$$

Dados dois inteiros positivos n e m e uma matriz quadrada $A_{n \times m}$, verificar se A é um quadrado mágico.

Atenção: o usuário deve fornecer os valores de n e m , e todos os elementos da matriz $A_{n \times m}$.

Tentaram me ajudar, mas mesmo assim o código não funcionou, se puder falar sobre o código antes da prova, ficarei grato!

```
#include <stdio.h>
```

```
int main ( ){
```

```
int Matriz[20][20], dg1, dg2, slinha, scoluna, linha , coluna , n , m, i , j;
```

```
printf("Digite a quantidade de linha da matriz (max 20): ");
```

```
scanf("%d", &n) ;
```

```
printf("Digite a quantidade de coluna da matriz (max 20): ");
```

```
scanf("%d", &m) ;
```

```
if(n == m){
```

```
    for(linha=0; linha<n ;linha++){
```

```
        for(coluna=0 ; coluna < m ; coluna++){
```

```
            printf("Defina o valor da Matriz na posicao[%d][%d] = " , linha , coluna);
```

```
            scanf("%f", &Matriz[linha][coluna]);
```

```
        }
```

```
    }
```

```
    for(i=0; i<n; i++){
```

```
        dg1 = dg1 + Matriz[i][i];
```

```
    }
```

```
    for(linha=0; linha<n; linha++){
```

```
        dg2 = dg2 + Matriz[linha][n-1-linha];
```

```
}  
for(i=0; i<coluna; i++){  
    scoluna += Matriz[i][0];  
}  
for(j=0; j<coluna; j++){  
    slinha += Matriz[0][j];  
}  
  
if((dg1 == dg2) && (slinha == dg2) && (scoluna == slinha) && (dg1 == scoluna)  
&& (slinha == dg1) && (dg2 == scoluna)){  
    printf("A matriz eh um cubo magico!");  
}  
else{  
    printf("A matriz nao eh um cubo magico!");  
}  
  
else{  
    printf("A Matriz nao eh quadrada");  
}  
  
return 0;  
}
```