

Lista de exercícios

Esta atividade prática pode ser feito em duplas e deverá ser entregue até o dia 01/07.

Estruturas homogêneas - strings

Escreva um programa em C para resolver os seguintes problemas com uso de vetores:

1. Faça um programa que peça para um usuário digitar o login e uma senha. O usuário poderá entrar no sistema somente se o login e a senha corresponderem. O login deve ser "admin" e a senha "1234". Se o login e a senha corresponderem, mostre uma mensagem aleatória para o usuário. Caso contrário, mostre uma mensagem de erro.
2. Faça uma aplicação que receba do usuário uma frase. Identifique o tamanho da frase e imprima esta frase tantas vezes quanto for a quantidade de caracteres. Ex: para a frase: "zé do povo", ela deverá aparecer 10 vezes na tela, pois tem 10 caracteres (incluindo espaços).
3. Crie um programa que recebe uma string e um caractere, e apague todas as ocorrências desses caractere na string.
4. Crie um programa que recebe uma string como entrada e duplique cada letra desta palavra. Por exemplo: se o usuário digitar a palavra "LONDRINA", a saída deverá ser "LLOONNDDRRRIINNA".
5. Júlio César usava um sistema de criptografia, agora conhecido como Cifra de César, que trocava cada letra pelo equivalente em duas posições à direita no alfabeto (por exemplo, 'A' vira 'C', 'R' vira 'T', etc.). Ao fim do alfabeto nós voltamos para o começo, isto é 'Y' vira 'A'. Nós podemos, é claro, tentar trocar as letras com quaisquer número de posições. A entrada contém vários casos de teste. A primeira linha da entrada contém uma string com até 50 caracteres maiúsculos ('A'-'Z'), que é a sentença após ela ter sido codificada através desta Cifra de César modificada. A segunda linha contém um número que varia de 0 a 25 e que representa quantas posições cada letra foi deslocada para a direita. Por exemplo, se a entrada da string for "LIPPSASVPH" e o número de posições for 4, a saída será "HELLOWORLD", pois cada caractere da entrada foi deslocado 4 posições antes.

Estruturas homogêneas - matrizes

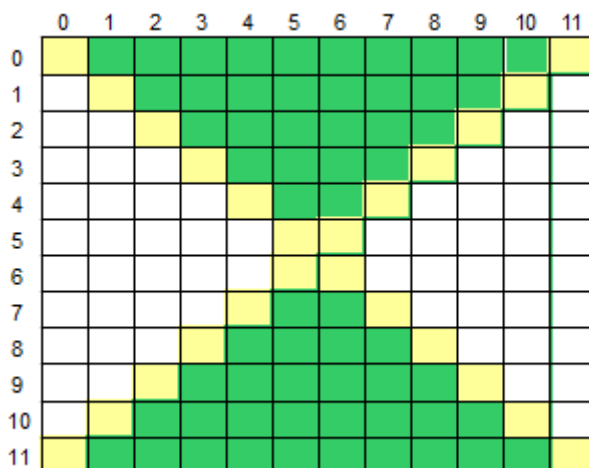
1. Faça um algoritmo que gere a seguinte matriz e imprima na tela:

```
1 1 1 1 1 1
1 2 2 2 2 1
1 2 3 3 2 1
1 2 3 3 2 1
1 2 2 2 2 1
1 1 1 1 1 1
```

2. Dada uma matriz A, leia uma matriz de ordem qualquer e imprima a quantidade de linhas e de colunas nulas da matriz. Uma linha ou coluna é dita como nula se todos os respectivos elementos são iguais a zero.
3. Dizemos que uma matriz inteira $A_{n \times n}$ é uma matriz de permutação se em cada linha e em cada coluna houver $n-1$ elementos nulos e um único elemento igual a 1. Dada uma matriz inteira $A_{n \times n}$, verificar se A é de permutação. A matriz abaixo por exemplo é uma matriz de permutação, pois satisfaz as propriedades acima:

```
0 1 0 0
0 0 1 0
1 0 0 0
0 0 0 1
```

4. Escreva um algoritmo que leia uma matriz 5x5 de elementos reais. Em seguida, o algoritmo deve calcular a soma dos elementos de cada linha e armazenar em um vetor de 5 posições. Cada posição deste vetor deve armazenar a soma da respectiva posição da linha, por exemplo, a posição 0 deve armazenar a soma dos elementos da linha 0, a posição 1 a soma dos elementos da linha 1 e assim por diante.
5. Escreva um programa em que o usuário deve ler um número inteiro N e o programa deve em seguida criar uma matriz $N \times N$ formado somente por 0. Em seguida, o programa deve substituir o valor 0 por 1 nas regiões destacadas conforme a figura abaixo. No final, o programa deve imprimir a matriz. O exemplo abaixo é para $N=12$.



6. O triângulo de Pascal é um triângulo numérico, cujos elementos são obtidos através da soma de dois elementos imediatamente acima e do antecessor do número de cima. Faça um algoritmo que leia um valor de N, representando a altura de um triângulo e que em seguida desenhe o triângulo de Pascal seguindo a lógica dos exemplos abaixo. Crie um programa que receba como entrada um valor N. Em seguida, o programa deve gerar a Torre de Pascal correspondente à matriz conforme o exemplo abaixo.

[illegible]

Estruturas heterogêneas - registros

1. **Escreva um programa em C que leia os dados de um tempo em horas, minutos e segundos e armazene em um registro (struct) chamado Hora, que contém os dados de horas, minutos e segundos, todos variáveis do tipo inteiro. O programa deve somente armazenar os dados no registro Hora caso o usuário entre com um tempo válido, isto é, o valor da hora deve ser um valor de 0 a 23 e os minutos e segundos valores de 0 a 59. Se o usuário digitar uma hora inválida, mostrar mensagem de erro e encerrar o programa. Em seguida, no programa, faça o seguinte:**

- a. Imprimir os dados do registro no formato hh:mm:ss.
- b. Calcular o tempo total das horas em segundos.

2. **Considere o seguinte modelo de registro que representa uma pessoa:**

- Nome (string)
- Data de nascimento (registro do tipo 'Data' – possui dia, mês e ano)
- CPF (string – somente dígitos numéricos)

Crie um registro do tipo acima. Os campos nome e CPF são do tipo string. A data de nascimento é um outro registro do tipo Data (dia, mês e ano) a ser definido também. Em seguida, crie um algoritmo para calcular a idade da pessoa de acordo com sua data de nascimento. Crie um algoritmo também para validar o CPF da pessoa (um CPF válido neste caso é se o seu tamanho é igual a 11 e se existem somente números na string).

3. **Considere o seguinte modelo de registro que representa uma pessoa:**

- Nome (string)
- Idade (int)
- Altura (float)

Crie um programa em que o usuário forneça dados de 5 pessoas de acordo com os dados acima e armazene em um vetor de registros. Logo em seguida, crie um algoritmo que percorra o vetor de registros e faça o seguinte:

- a) Imprima os dados de todas as pessoas que tem mais de 18 anos.
- b) Calcule e imprima a média da altura de todas as pessoas.

4. **Para o controle de veículos que circulam em uma determinada cidade, a Secretaria dos Transportes criou um registro padrão de veículos com as seguintes informações:**

- Proprietário
- Combustível (string - álcool, diesel ou gasolina)
- Modelo (string)
- Cor (string)
- Número de chassi (inteiro)
- Ano (inteiro)
- Placa (3 primeiros valores alfabéticos e os 4 restantes números)

Construa um algoritmo que leia uma quantidade arbitrária de veículos, que pode ser definido pelo usuário. Em seguida:

- a) Liste todos os proprietários cujos carros são do ano de 1980 ou posterior e que sejam movidos a diesel.
- b) Liste todos os proprietários cujos carros são movidos a álcool ou a gasolina e que sejam do ano de 2000 para cima.
- c) Liste todos os veículos cujas placas comecem com a letra 'A' e que o último dígito da placa seja um número par.