30/09/2019

Lista de exercícios
LP1-ADS

Prof. Luciano Bernardes de Paula

1 – Escreva um programa que receba nove valores, armazene-os em uma matriz e, em seguida, os apresente na tela. Use uma matriz 3 x 3.

2 - Faça um programa que utilize uma matriz TAM x TAM e preencha-a com zeros e em seguida preencha as duas diagonais com o valor 1. Após isso, imprima a matriz resultante. Utilize TAM como uma constante e teste o programa com TAM = 10, 20, 25 e 30.

3 - Faça um programa que utilize uma matriz TAM x TAM e preencha-a com zeros e em seguida preencha somente as linhas e colunas que estiverem na "borda" com o valor 1. Após isso, imprima a matriz resultante. Utilize TAM como uma constante e teste o programa com TAM = 10, 20, 25 e 30.

4 - Faça um programa que utilize uma matriz TAM x TAM e preencha-a valores aleatórios entre 0 e 500. Imprima a matriz resultante e em seguida são apresentadas duas opções para o usuário: a primeira opção faz com que seja apresentado o menor valor presente na matriz e a segunda opção o maior. Utilize TAM como uma constante e teste o programa com TAM = 10, 20, 25 e 30.
Dicas:
- para gerar números aleatórios, use a função *rand( )*. Para usá-la é preciso incluir a biblioteca ***stdlib.h***.
- para usar a função *rand( )* use:

variável = rand();

- para que sejam gerados números entre 0 e n, é possível calcular o resultado do resto da divisão do resultado da função *rand()* por n+1. Exemplo:
a = rand() % 101; // gera valores entre 0 e 100
- toda vez que a função rand() é executada, ela gera um número "aleatório" de acordo com um valor chamado semente, que é próprio da função. Para cada programa, a semente será uma. Para inicializar a semente para cada execução de programa, utilize a função *srand( )*. Essa função recebe um valor que será usado como semente. Uma forma de gerar sementes diferentes para cada execução é passar o horário do computador para que seja usado como semente. Cada vez que for executado, a probabilidade de ser utilizada uma semente diferente é grande. Para passar o horário do computador como semente, utilize:

srand(time(NULL)): // use esse comando antes de usar a função rand( ) no seu programa

Para utilizar a função ***time()***, é preciso incluir a biblioteca ***time.h***.

5 - Faça um programa que utilize uma matriz TAM x TAM e preencha-a valores aleatórios entre 0 e 500. Em seguida o programa deve ordenar a matriz em ordem

crescente (da esquerda para direita, de cima para baixo) e apresentá-la na tela. Utilize TAM como uma constante e teste o programa com TAM = 10, 20, 25 e 30.

6 – Faça um programa que controle, de maneira simples, a conta de mesas em um restaurante. O programa deve apresentar o seguinte menu:

Menu
0 – Registrar valor em uma mesa
1 – Ver conta de uma mesa
2 – Pagar conta de uma mesa
3 – Sair

As mesas são representadas em uma matriz 10 x 10 de floats. Cada posição da matriz armazena o valor da conta da mesa. Todos as mesas começam com saldo 0,0 e, conforme o usuário escolhe a opção 0, será adicionado um valor à conta. Cada mesa é reconhecida pelo sistema por um par de valores: linha e coluna. A única opção que finalizar o programa é a 3, todas as outras devem, após sua execução, voltar ao menu. Segue abaixo o que cada opção deve fazer (e como deve ser feito).

- Ao escolher a opção 0, o programa deve pedir para entrar com os valores da linha e coluna, depois pedir o valor a ser adicionado na conta e fazer essa atualização;

- Ao escolher a opção 1, o programa deve pedir deve pedir para entrar com os valores da linha e coluna e apresentar o valor da conta atual daquela mesa;

- Ao escolher a opção 2, o programa deve pedir a entrada da linha e coluna, depois do valor pago e fazer o cálculo se deve haver troco ou não;

- Ao escolher a opção 3 o programa deve checar se não há nenhuma mesa em aberto (com conta a ser paga), caso isso seja verdadeiro, o programa finaliza. Caso haja mesa(s) em aberto, o programa deve avisar esse fato, informando quais mesas possuem conta aberta e não deve fechar, devendo voltar ao menu.