

Trabalho de Programação

1. Faça uma matriz 8x8 e utilize 3 elementos dela, de forma que sejam adjacentes e que representem um ângulo de 90 graus. Preencha essa matriz com zeros, e apenas estes 3 elementos devem conter outro caracter. O movimento deve sempre ser adjacente. Você deve utilizar uma tecla para indicar o movimento (que você deve escolher - WASD), e cada vez que essa tecla for clicada, um movimento de mudança deve acontecer. Você deve fixar uma das extremidades, e movimentar a outra. Por exemplo, se eu usar a tecla de movimentação, um dos lados dela precisará trocar de lugar (você define como essa troca acontece). Você não pode ultrapassar os limites da matriz: quando chegar nas extremidades, faça novas trocas retornando para o interior da matriz. Utilize uma tecla para finalizar o programa (ESC). A cada movimento, você deve mostrar a situação da matriz.

2. Faça uma matriz 8 x 8 e inicialize ela com zeros. Em seguida, crie um quadrado de valores qualquer, que ocupe 4 posições. Faça um programa que percorra a matriz com este quadrado, andando sempre para uma posição adjacente (cada movimento somente poderá acontecer após o usuário digitar uma das teclas previamente definidas, que você deverá escolher: para cima, para baixo, para a direita e para a esquerda). Quando terminar uma linha, e você continua digitando para andar para o lado, o seu quadrado deve passar para a próxima linha. E isso deve acontecer até que o usuário digite uma tecla para finalizar (que você irá definir).

3. Faça um programa com uma matriz [8][8] que preencha cada uma das diagonais dela de forma consecutiva e automatizada, utilizando um valor único para cada diagonal, e que faça com que após o preenchimento, os valores sejam atualizados para o valor da próxima diagonal. Escolha e informe ao usuário uma tecla que será utilizada para a finalização (enquanto o usuário não finalizar, continua andando com as diagonais). Por exemplo, em uma matriz 4x4:

1 2 3 4		2 3 4 1
2 3 4 1	→	3 4 1 2
3 4 1 2		4 1 2 3
4 1 2 3		1 2 3 4

4. Faça um programa que desenhe uma matriz 8 x 8, e que preencha ela com zeros. Apenas a primeira posição deverá conter um traço. De acordo com a utilização do teclado, da forma que você achar mais conveniente, escolha 4 teclas para você mover o traço uma posição, respectivamente, para esquerda, para cima, para a direita e para baixo. Quando o movimento extrapolar os limites da matriz, dê a volta nela e inicie pela outra extremidade. Você deve contar quantas interações o usuário realizou, e escolher uma tecla para finalizar o programa, que deverá ser informada inicialmente para o usuário.

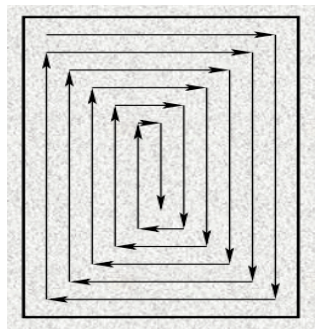
5. Faça um programa que desenhe uma matriz 8 x 8, e que preencha ela com zeros. Apenas a primeira posição deverá conter um traço. De acordo com a utilização do teclado (escolha 4 teclas que achar conveniente) você deverá mover o traço uma posição, respectivamente, para esquerda, para cima, para a direita e para baixo. Você não pode ultrapassar os limites da matriz. Quando atingir a borda, por exemplo, o usuário deverá digitar os comandos que façam a traço andar para qualquer

outro lado, mas não poderá sair dos limites da matriz.

6. Você conhece o diagrama de Pauling? Pois então, faça um programa que realize o movimento similar ao do diagrama de Pauling, mas agora em uma matriz 8 x 8. Inicialize a matriz com zeros, e em seguida você deverá percorrer toda a matriz, de forma automática, iniciando da posição [7][0], utilizando apenas um caractere diferente para esse trajeto. Na verdade, você deverá cruzar todas as diagonais da matriz, uma posição por vez. Quando chegar na posição [0][7] você deverá continuar iniciando o trajeto da posição [7][0]. O programa não pode parar.

7. Crie uma matriz 8 x 8 de forma que exista apenas um único caminho livre do ponto MAT[0][0] até o ponto MAT[7][7]. O seu programa deve ser capaz de identificar qualquer caminho livre para chegar até o destino final. O caminho será planejado na inicialização da matriz, e deve funcionar para qualquer caso (ou seja, se a inicialização for diferente, deverá funcionar). Considere que o caractere traço (-) indica um caminho livre, e que o caractere Zero (0) é utilizado naquelas posições bloqueadas. Apresente um algoritmo que seja capaz de mostrar a movimentação, do início ao fim, de um elemento (um caractere qualquer percorrendo a matriz, diferente de traço e zero) percorrendo a matriz até chegar ao seu ponto final (Mat[7][7]). Para cada passo dado pelo programa, selecione uma tecla para realizá-lo. Por exemplo, serão 16 passos até chegar ao final, então você irá clicar neste tecla 16 vezes.

8. Faça uma matriz 8 x 8 e preencha ela inicialmente com zeros. Altere os valores da matriz de acordo com a imagem abaixo:



Ou seja, na primeira iteração, você altera os valores da primeira linha (completa). Na segunda iteração, você altera os valores da última coluna. Na terceira iteração, você altera a última linha, e na quarta iteração, você altera a primeira coluna, exceto a primeira linha. Após fechar um “quadrado de alterações”, você recomeça as alterações no quadrado interno, até que toda a matriz seja percorrida, até o seu interior. Cada linha ou coluna será alterada após digitar um caractere previamente estabelecido (você pode escolher) e o programa somente encerrará quando chegar ao interior da matriz. A cada linha ou coluna alterada, você deve apresentar a matriz.

9. Faça uma matriz 8 x 8 e inicialize-a com zeros. Em seguida, altere ordenadamente e de forma crescente, os valores da primeira linha. Depois disso, vá para a segunda linha, mas continue a preencher sua matriz ordenadamente, mas a contar da última coluna, em direção a primeira coluna. Ao chegar na primeira coluna, vá para a terceira linha, e continue assim consecutivamente, até que a matriz esteja completamente preenchida. Apresente a matriz a cada linha que for alterada. Quando você finalizar o preenchimento das colunas, repita o procedimento, mas agora preenchendo as colunas, subindo uma coluna, e descendo na outra. Por exemplo:

```

1 2 3 4      →   21   16   15
8 7 6 5      20   17   14
9 10 11 12    19   18   13

```

10. (1,0 Extra) Faça um jogo similar ao jogo de dominó. Você deve iniciar o jogo com a peça (0,0), e então tentar encaixar as peças ao máximo possível (considere as 28 peças do dominó). Para cada peça, você pode definir um código para escolhê-la na tela (pode ser a, b, c, d...), e sinalizar quando ela já foi utilizada. Quando uma peça já foi utilizada, ela não poderá mais ser utilizada. Enquanto houverem peças que possam ser encaixadas, o jogo continua solicitando ao usuário que encaixe uma das peças disponíveis (seu programa deve fazer esse controle). Você deve sempre apresentar ao usuário como as peças já foram encaixadas, e quais as peças ainda estão disponíveis (ou seja, a cada execução, mostra as tabelas abaixo). O jogo finaliza quando não há mais peças que possam ser encaixadas. Você pode utilizar uma matriz para representar as 28 peças, com uma coluna extra que indica quando a peça já foi utilizada, e uma outra matriz que indica a ordem que as peças foram conectadas. Por exemplo:

Valor1	Valor2	Usada		Valor1	Valor2
0	0	1		0	0
0	1	0		0	1
0	2	1			
0	3	0			

Você pode conectar o Valor2 em Valor 1 (tenha cuidado, pois isso inverte a ordem de escrita na segunda tabela). A cada execução do jogo, você deve mostrar as duas tabelas. A ordem de conexão será somente em um sentido, ou seja, depois de conectar a peça (0,1) você deverá obrigatoriamente conectar uma peça (1, X). Enquanto houverem peças para se conectar na última peça escolhida, o jogo não pode finalizar.

11. (1,0 Extra) Jogo de Damas

12. (1,0 Extra) Jogo Batalha Naval

13. (1,0 Extra) 2048 em matriz 8x8