Tiago Augusto Simionato Tozo

Somamax:

Primeiro o programa recebe um número n como entrada e um vetor de tamanho n é criado, então o programa recebe n números e os coloca dentro do vetor através de um for(i=0;i<n;i++) onde o índice do vetor é o próprio i.

Para fazer o cálculo da soma max é aberto um for de j=0 até j<n, e a cada interação dele a auxiliar de soma (auxsoma) é zerada para não atrapalhar a próxima conta. Dentro desse for um outro for de i=0 até (i+j)<n onde a auxsoma é usada para calcular a soma de todos os termos do vetor até o fim. Dentro do segundo for um if compara se a auxsoma atual é maior do que a soma até o momento e se esse for o caso soma passa a ter o valor de auxsoma e os índices j e i+j são guardados como os índices para serem escritos na resposta com um +1, já que no vetor os índices vão de 0 até n-1 e não de 1 a n.

O for (i+j)<n dentro do for j=0 até j<n serve para que conforme o for de fora é executado o índice inicial da conta (i=0) seja o próprio j, por isso o vetor também é vet[i+j].

Por último um if vê se a soma = 0 (todos os números são negativos, devido ao fato de soma começar com o valor 0(soma<auxsoma)), se esse for o caso um printf escreve o próprio valor da soma(0), se não, é escrita a soma e os índices.

Qmagico:

Quadrados mágicos ímpares – Como a lógica para um quadrado mágico ímpar é a mesma independente do lado do quadrado, o meu programa usa uma função “void QuadradoMagicoImpar(int n)” que recebe um inteiro(que deve ser ímpar) e calcula seu quadrado mágico.

Para calcular o quadrado mágico de lado n eu usei uma técnica que é colocar o número 1 na primeira linha e na coluna do meio {vet[0][n/2]}, e ir subindo o valor de 1 em 1 até n^2 colocando o próximo número sempre uma linha para cima e uma coluna para a direita, e sempre que um valor “sai” do quadrado ele é levado para o outro lado, então se um número x está na linha 1 coluna n-1 o x+1 será colocado na posição linha n e coluna n, além disso se após um movimento (cima e direita) a casa já estiver ocupada o próximo número vai embaixo (mesma coluna, linha debaixo) do numero anterior.

Para isso eu usei um for(x=2;x<=n*n;x++) para subir o valor de 1 em 1, e a cada interação k=k-1 e l=l+1, para depois igualar vet[k][l]=x. Dentro desse for tem 3 ifs, um iguala ‘k’ a ‘n’ se ‘k’ for igual a -1 e outra iguala l a 0 se l = 3, para o quadrado não “estourar”. O outro if confere se uma casa já está ocupada, para isso toda a matriz é igualada a 0 antes do for, o if confere se o valor na casa é diferente de zero, já que ele só muda se um valor já foi colocado nessa casa, se o valor for diferente de zero então o k é igualado a j(valor inalterado da posição atual)+1 e o l a i(valor inalterado da posição atual).Depois dos 3 ifs o programa faz a atribuição vet[k][l]=x e iguala o ‘j’ e o ‘i‘ a ‘k’ e ‘l’ respectivamente, para que o j e i estejam corretos para o 3ºif na próxima interação.

Quadrado Mágico n=4

Para quadrados mágicos de lado n divisível por 4 a lógica é outra, primeiro se divide ele em quadrados de tamanho n/4 nas extremidades e um quadrado n/2 no centro, ai se faz um “loop” de 1 a n*n somando de um em um seguindo a ordem do quadrado, indo de direita para esquerda e de cima para baixo, colocando os valores nas casas somente se a casa fizer parte das selecionadas anteriormente. Após isso, se terá um quadrado

1 X X 4

X 6 7 X /*QUADRADO 4X4*/

X 10 11 X

13 X X 16

Para terminar o quadrado é só fazer outro “loop” seguindo a mesma lógica mas dessa vez indo de 16 até 1 diminuindo de um em um e colocando os valores nas casas livres.

Em um quadrado magico 4x4 os quadrados menores acabam sendo por coincidência a diagonal principal e secundaria, tendo isso em mente eu igualei elas a 17(valor que nunca aparecera dentre as opções (1 a 4^2) e o resto do quadrado foi igualado a 0. Ai fiz um for de subida (x=1, x=x+1 dentro de um par de fores que percorrem a matriz) e usei um if para ele somente igualar os valores se estiver entre os quadrados selecionados (vet[j][i]==17). E por último fiz outro par de fores que percorrem a matriz, dessa vez indo de n^2 a 1 subtraindo de um em um e usando um if para só atribuir o valor se a casa for uma parte do vetor não preenchida (vet[j][i]==0).