

Documentação: Trabalho de Matematica Discreta.

Soma máxima:

Primeiro o programa pede o tamanho do vetor, no caso a quantidade de números que seram somados. E com um for, fará o preenchimento de todo o vetor.

```
/*    scanf("%d",&quantidade);
    int sequencia[quantidade];
    for(D = 0 ; D < quantidade; D++)
    {
        scanf("%d",&sequencia[D]);
    } */
```

Para achar a maior soma entre um intervalo, meu programa vai avaliar cada elemento do meu vetor, depois vai comparar a soma de dois elementos com o valor de cada elemento, depois a soma de três em três, quatro em quatro, fazendo a comparação de todas as somas até N (o final do vetor) e vai guardar o maior valor da soma na variável TOTAL.

```
/*    for(j = 0; j < quantidade; j++){
        for(k = 0; k < quantidade; k++){
            TOTAL =0;
            for(i = j; i < quantidade-k; i++){

                TOTAL = TOTAL + sequencia[i];

            }
        }
    }
```

Meu programa vai fazer isso durante todo o vetor. E para saber em qual intervalo ele faz a maior soma, criei a condição de que onde ele achar a maior soma, ele vai printar o valor do “i” (que é o primeiro valor do intervalo) e o valor do “j” (é o ultimo elemento do intervalo).

```
/*    if(TOTAL > soma){
        soma = TOTAL;
        AUXj = j;
        AUXi = (i-1);} */
```

obs.: Essa subtração feita com AUXi, é realizada para não ultrapassar o vetor.

Quadrado Mágico:

O programa primeiro vai ler o tamanho da matriz, uma vez que ela é quadrada, o programa so vai precisar de um parametro para saber como a matriz vai ser.

```
/* scanf("%d", &N);*/
```

Se ele colocar um valor 3, será impresso uma matriz tipo a do exemplo a seguir;

0 0 0

```

0      0      0
0      0      0

```

uma matriz 3 x 3.

Após passado o parâmetro do quadrado, o programa irá avaliar se o quadrado é ímpar ou se é par.

Sendo o quadrado do tipo par. O preenchimento sera feito da seguinte forma:

1º- Preenche a diagonal principal do quadrado;

```

/*      if (coluna == linha) {

                quadrado[coluna][linha] = contador; } */

```

2º- Preenche a diagonal secundaria;

```

/*      else if (coluna == (N - 1 - linha)) {

```

```

                quadrado[coluna][linha] = contador; } */

```

3º- Após feito o preenchimento da diagonal principal e secundária (de forma crescente), depois o programa ira preencher os espaços vazios da matriz com os valores restantes da sequencia de forma decrescente.

```

/*      else if (quadrado[coluna][linha] == 0) {

```

```

                quadrado[coluna][linha] = contador2 } */

```

Obs.: Essa logica so ira funcionar para matriz do tipo N=4. O trecho do programa abaixo faz o que foi dito no terceiro item:

```

        contador = 0;
        int contador2 = 17;
        for (coluna = 0; coluna < N; coluna++) {
            for (linha = 0; linha < N; linha++) {
                /**acrescentar os valores no quadrado, preenchendo primeiro a diagonal principal e secundaria, depois
colocando os outros
valores em ordem decrescente.**/
                contador++;
                contador2--;
            }
        }

```

A matriz sendo do tipo ímpar, usei a logica de Loubere, um diplomata francês. Segue abaixo a lógica dele :

Um método para a construção de quadrados mágicos de ordem ímpar foi publicado pelo diplomata francês da Loubère em seu livro, uma nova relação histórica do reino de Sião (Du Royaume de Siam, 1693), no capítulo intitulado O problema da praça mágica de acordo com os índios. O método funciona da seguinte forma:

O método prescreve a partir da coluna central da primeira linha com o número 1. Depois disso, o movimento fundamental para preencher os quadrados está diagonalmente para cima e para a direita, um passo de cada vez. Se um quadrado cheio for encontrado, um se move verticalmente para baixo de um

quadrado em vez disso, então continua como antes. Quando um movimento "para cima e para a direita" deixaria o quadrado, ele é enrolado em torno da última linha ou primeira coluna, respectivamente.

Partindo deste raciocínio, mudei o parâmetro de preenchimento. Partindo que o número 1 sempre vai estar na linha do meio da primeira coluna, o movimento realizado para preencher os próximos quadrados é realizado indo para baixa e para esquerda, caso encontre um número, volta para a origem e depois preenche na casa a direita da origem.

Segue abaixo o trecho do programa que realiza esses movimentos:

/*

```
while (stop > 1) {
    stop--;
                                contador++;

    linhaanterior = linha;
    colunaaanterior = coluna;

    linha--;
    coluna++;

    if (linha < 0) {
        linha = N-1;
    }

    if (coluna > (N - 1)) {
        coluna = 0;
    }

    if (quadrado[linha][coluna] == 0) {

        quadrado[linha][coluna] = contador;

    } else {
        coluna = colunaaanterior;
        linha = linhaanterior;
        linha++;

        if (linha >= N) {
            linha = 0;
        }

        quadrado[linha][coluna] = contador;
    }
}
```

} */

