

Documentação: Trabalho Prático – Matemática Discreta

Soma máxima:

O exercício da soma máxima consiste em, dada uma determinada sequência de números, determinar qual a maior soma possível dos números em sequência.

Para isso poderia ser feito a soma de todas as sequências possíveis e guardado qual o maior valor de todas essas somas.

Dessa forma o código somamax.c consegue dizer qual o valor da maior soma.

```
int main(int argc, char *argv[]) {
    int n, aux, aux2, aux3;
    int soma, Msoma, indiceI, indiceF;
    while(n<3||n>20){          //Garante que a sequência fique entre 3 e 20
        scanf ("%d", &n);
        if(n<3||n>20)printf("tamanho de sequencia invalido, entre com um numero
de 3 a 20\n");
    }
    int vetor[n];
```

Esse trecho do código é responsável pela declaração das variáveis e pela inicialização do vetor que terá os valores da sequência.

A entrada do número “n” deve ser de um número de 3 até 20, que será responsável por determinar o tamanho do arranjo “vetor [n]”.

```
        for (aux=0; aux<n; aux++) scanf("%d", &vetor[aux]);    //entrada de cada
valor do vetor
```

O laço é responsável para dar a entrada de cada valor do vetor, podendo ser um número negativo, positivo ou nulo.

```

for (aux=0; aux<n; aux++){
    for (aux2=0; aux2<n; aux2++){
        soma=0;
        for (aux3=aux; aux3<n-aux2; aux3++) soma+=vetor[aux3];
        if (Msoma<soma){
            Msoma=soma;
            indiceI=aux+1;
            indiceF=n-aux2;
        }
    }
}

```

Esse trecho em questão é responsável por fazer a soma de cada sequência, indo dos índices auxiliares “aux” até o “n – aux2” e guardando a soma que tiver o maior resultado. Dessa forma a cada novo ciclo desse laço ele vai até um índice uma unidade menor e comparar com a maior soma até então, salvando o valor da soma, o índice inicial e o índice final da melhor sequência. O primeiro laço muda o valor do índice inicial, o segundo, do índice final e o terceiro faz a soma dos valores do índice inicial até o índice final.

```

if (Msoma>=0){
    printf("Soma: %d\n", Msoma);
    printf("Indices: %d a %d", indiceI, indiceF);
}
else {
    printf("Soma: 0\n");
    printf("Indices: 0 a 0");
}
return 0;
}

```

A parte final do código é responsável por dar a saída do programa, mostrando qual a maior soma e os índices inicial e final. Caso a maior soma for negativa, a maior soma é igual a 0.

Quadrado mágico:

O exercício do quadrado mágico consiste em gerar um quadrado $N \times N$ numerados de 1 até N^2 e que a organização dos números seja tal que a soma dos números das linhas seja igual à soma das colunas e igual à soma das diagonais.

Para quadrados de N ímpar, existe um certo padrão de organização. Colocando o valor inicial (1), na posição do meio da primeira linha, o próximo número deve ser colocado uma linha para cima e uma coluna para a esquerda, caso já tenha sido colocado um valor ali, coloque esse número uma linha abaixo da linha destino.

Para o quadrado de $N = 4$, comparando uma matriz com valores crescentes e outra com valores decrescentes com uma matriz organizada da maneira :

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

As posições com o número 1 recebem o valor da matriz crescente e as posições com valor 0, as da matriz decrescente.

```
int main(int argc, char *argv[]) {
    int n, n2, aux, aux2;
    do{
        scanf("%d", &n);
        if (n<3||n>5) printf("Valor invalido para n, entre com um numero entre 3 e
5\n");}
        while (n<3||n>5);
    int i=0, j=n/2, k=0;
    int SomaMagica=n*((n*n+1)/2), soma=0;
    int quadrado[n][n];
    int organizado[n][n];
```

```
int comparacao[4][4]={1,0,0,1,0,1,1,0,0,1,1,0,1,0,0,1};
```

O início do código é responsável por declarar algumas variáveis, ler o valor da entrada para “N” e garantir que ele esteja entre 3 e 5.

```
for (aux=0; aux<n; aux++){
    for (aux2=0; aux2<n; aux2++){
        quadrado[aux][aux2]=n*aux+1+aux2;
    }
    for (aux=0; aux<n; aux++){
        for (aux2=0; aux2<n; aux2++){
            organizado [aux][aux2]=0;
        }
    }
}
```

Esse laço coloca os valores para as matrizes, sendo uma inicializadas todas em 0 e a outra com valores de 1 ate N^2 de forma ordenada.

```
if(n%2){
    for(aux=0; aux<n; aux++){
        for(aux2=0; aux2<n; aux2++){
            if (organizado[i][j])i++;
            organizado[i][j]=quadrado[aux][aux2];
            i--;
            j++;
            if (i<0) i=n-1;
            if (j==n) j=0;
        }
    }
}
else {
    for (aux=0; aux<n; aux++){
        for (aux2=0; aux2<n; aux2++){
            if (comparacao[aux][aux2])organizado[aux][aux2]=quadrado[aux]
[aux2];
            else organizado[aux][aux2]=quadrado[n-1-aux][n-1-aux2];
        }
    }
}
```

Essa condição é responsável por gerar os quadrados ímpares, colocando os números em locais determinados pelo padrão descrito no início, em que a linha “i” vai subindo e a coluna “j” indo para a esquerda, se chegar até o final da matriz, ela vai para o lado oposto e se já tiver um valor alocado, ela coloca na linha abaixo.

O quadrado de $N=4$ é gerado pela comparação entre os valores 1 e 0 para saber qual valor deve ser colocado, lendo a matriz de forma crescente ou decrescente.

```

    for (aux=0; aux<n; aux++) soma+=organizado[0][aux];
    //if (soma==SomaMagica)printf("Ok");
    printf("n = %d, Soma = %d\n", n, soma);
    for (aux=0; aux<n; aux++){
        for (aux2=0; aux2<n; aux2++)
            printf("%d ", organizado[aux][aux2]);
        printf ("\n");}

    return 0;
}

```

O final do código é responsável por dar a saída do programa, apresentando qual o valor da soma e qual é o quadrado gerado.