

Documentação - Quadrado Mágico

Primeiro começo o programa com dois loopings de repetição em que serve para o usuário inserir os elementos da matriz. Inicialmente eu havia feito da seguinte forma:

C/C++

```
for(int i = 0; i < n; i++){  
  
    for(int j = 0; j < n; j++){  
  
        printf("Digite o elemento (%d,%d) da matriz:",i,j);  
        scanf("%d",&matriz[i][j]);  
    }  
}
```

Somente depois que eu tentei pensar em algumas alternativas para colocar os elementos em apenas uma linha de input. Neste caso acabei optando por usar o `getchar()`, antes do `scanf` para de alguma forma limpar o espaço deixado entre os elementos.

Depois da matriz declarada era necessário os cálculos necessários para verificar o valor da soma das linhas, soma das colunas, soma da diagonal principal e soma da diagonal secundária.

Eu declarei posteriormente dois vetores, de tamanho `n` (escolhido pelo usuário anteriormente), em que seria armazenado os valores de soma linhas e soma coluna. Enquanto as somas das diagonais vão ser “constantes”.

Em todos os cálculos foi necessário utilizar loopings de repetição para percorrer os elementos da matriz utilizando de maneira inteligente seus indexadores. Por exemplo:

C/C++

```
int vetor_soma_linhas[n];  
  
for(int i = 0; i < n; i++){  
  
    soma_linhas = 0;  
  
    for(int j = 0; j < n; j++){  
  
        soma_linhas = soma_linhas + matriz[i][j];  
    }  
    vetor_soma_linhas[i] = soma_linhas;  
}
```

Quando o looping interno acaba, significa que a primeira linha acabou, então ele armazena essa soma no vetor soma linhas [i], e vai para a próxima linha, sendo assim necessário reiniciar a soma linha para zero, para começar tudo novamente. E de maneira análoga foi feito o cálculo da soma de colunas, apenas invertendo os loopings i e j, o que estava fora passou para dentro e vice e versa.

Já na parte do cálculo das diagonais foi diferente. A diagonal principal foi tranquila de fazer, apesar de utilizar um método em que posteriormente acabei percebendo (por buscas na internet sobre diferentes métodos de resolução do exercício) não ser o mais eficiente. Porém optei por deixar ele mesmo pois queria deixar o código mais autoral possível. A maneira mais eficiente seria:

C/C++

```
for (i = 0; i < n; i++) {  
    soma_diagonal_principal += matriz[i][i];  
}
```

Já a diagonal secundária achei mais difícil, tendo que recorrer a pesquisas na internet e verificar formas diferentes de resolução. Depois de verificar essa resolução vi que não seria um trecho de código que eu conseguiria (pelo menos nesse momento) desenvolver sozinho. Resumindo ...a lógica nos indicadores (índices) foi algo complicado.

C/C++

```
for(int i = 0; i < n; i++){  
  
    soma_diagonal_secundaria+=matriz[i][n-i-1];  
}
```

Depois de ter vetores e valores absolutos tive apenas que comparar esses valores utilizando if e & & para que as condições do quadrado mágico fossem estabelecidas. Depois disso utilizei uma lógica desenvolvida por mim mesmo para caso esses valores fossem todos iguais. Pois caso as condições do if sejam todas atendidas (significando que o quadrado mágico é verdadeiro) a soma será sempre igual ao tamanho n. Depois apenas utilizei a condição para imprimir a matriz caso seja um quadrado mágico, e caso não seja imprimir resultado =-1.