

LÓGICA DO PROGRAMA

A lógica de uma espiral quadrada baseasse em uma certa alternância entre incrementos e decrementos em "x" ou "y". Basicamente resume-se em, a partir da origem do eixo xy, incrementa-se uma vez y, depois decrementa-se uma vez em x, após isso decrementa-se duas vezes em y e incrementa-se duas vezes em x e assim sucessivamente. Analisando o exemplo citado anteriormente percebe-se q ora executa-se y e ora x, alternando a vez entre si. Além de alternar entre x e y, alterna-se também entre decretação e incrementação, sempre aumentando de um em um a quantidade.

Exemplo

o programa começa com:

+y -x -y -y +x +x +y +y +y -x -x -x

Separando-se x de y temos:

+y -y -y +y +y +y -y -y -y -y || -x +x +x -x -x -x +x +x +x +x

Ao analisar as vezes que a incrementação e decretação ocorrem:

+1y -2y +3y -4y || -1x +2x -3x +4x

Podemos separar também as incrementações das decrementações: (usaremos o exemplo de apenas uma das variáveis, pois a lógica para ambas são similares)

-1x -3x -5x -7x || +2x +4x +6x +8x

O PROGRAMA

2- (**variáveis**) Entramos com 8 variáveis (a, d, f, g, h, x, y, n)

3- (**valores e funções das variáveis**) Igualamos "x" e "y" a zero, uma vez que as primeiras coordenadas do nosso programa são (0,0), pois é de lá que a contagem começa. Igualamos também "d", "f" a 1, pois cada vez que uma determinada função se repetir haverá uma incrementação neles, no caso de "d" e "f", serão responsáveis para definir se "x" ou "y" será incrementado ou decrementado, no caso de "h" ele será um contador que acompanhará a repetição do programa.

4- (**scanf e "n"**) O programa irá escanear a posição desejada por quem executar o programa e atribui-la à variável "n", a qual será responsável de fazer o programa repetir "n" vezes.

5- (**primeiro for**) O programa executa uma função "for", iniciando com "a" igual a 1, repetindo-se "n" vezes. Logo ao entrar no "for" existe um "if else" responsável por direcionar a execução do programa para x ou y. Como a espiral quadrada inicia-se com uma alteração no valor de y, definiremos que a primeira execução sempre tem que ser em y.

6- (**primeiro if else**) foi definido no programa um "if (a%2==0)" com a função de, quando "a" for ímpar o programa entrará no else e alterará o valor da variável y, quando "a" for par entrará no if e alterará o valor de x, como no final da execução do for há uma atribuição ao valor de a e, caso a ainda seja menor ou igual a n, o programa executará novamente com o valor de a alterado, fazendo com que sempre alterne entre x e y, assim como é a lógica da espiral quadrada.

7- (**segundo if else**) Ao entrar nesse if ou else, o programa apresenta outro if else com o mesmo objetivo do primeiro, porém no lugar de segregar x e y, esse if else tem a função de segregar

incrementação de decretação. No caso do x, encontramos um "if(d%2==0)", foi definido acima que "d" iniciava-se no programa com o valor igual a 1. Como ora decreta x e ora incrementa, e como no final do if else o programa faz uma incrementação no valor de "d", cada vez que o programa executar esse if else, "d" terá um valor diferente, alternando assim entre incrementação e decretação. É usado exatamente a mesma linha de raciocínio para tratar o if else no y.

8- **(segundo for)** Assim que o programa entra no segundo if else, há um for, no caso do if else de x tem-se "for (g=0; g<d; g++)". O objetivo desse for é realizar a incrementação ou decretação "d" vezes, como vimos no 3 exemplo da lógica do programa, a incrementação e decretação se realiza inicialmente 1 vez, depois 2, em seguida 3, após isso 4 e sucessivamente, por isso o for irá se repetir d vezes, a princípio d é igual a 1, na próxima vez que o for será executado, d já assumirá o valor de 2 e assim sucessivamente, fazendo com que ele se repita a quantidade de vezes necessárias para realizar a incrementação ou decretação com sucesso.

9- **(if dentro do segundo for)** visa justamente encerrar a repetição do programa quando h assumir um valor igual a n. Vimos que h é uma variável que acompanha a repetição do programa, ou seja, quando o programa se executar n vezes, h será igual a n, satisfaremos as condições desse if, fazendo com que ele se execute, imprimindo as coordenadas desejadas e retornando o programa a 0.