

Universidad Rafael Landívar

Microprogramación

Laboratorio

Ingeniera Karen Liska

Proyecto Uno - Espiral Cuadrada

Mario Roldan

Héctor Zetino

Guatemala 15 de octubre 2021

Contenido

Objetivos3

Análisis del problema4

Metodología de trabajo5

Código.....6

Imágenes del programa8

Puntos de mejora 10

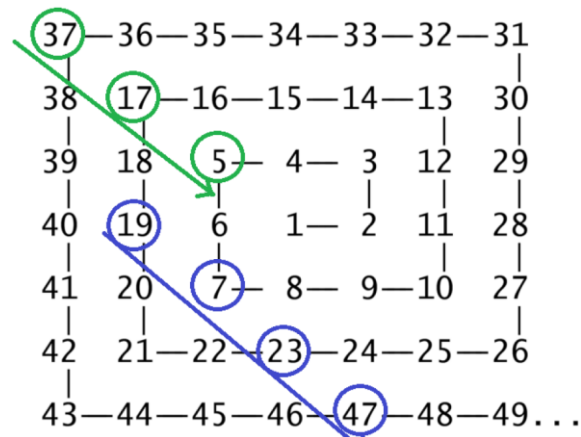
Conclusión 11

Objetivos

1. Ingresar numero valido y generar listado de números
2. Listar números, coordenada dentro del plano cartesiano
3. Ubicar "1" como (0,0)

Análisis del problema

En el presente proyecto se plantea la búsqueda de números primos, por medio de una matriz de números, de 1 a n, dado que 1 sea el centro y se realice una espiral cuadrada alrededor del mismo.



Tomando el primer paso lógico como 1 (0,0), 2 (1,0), 3(1,1), 4(0,1) y así con su consecuencia, esto para realizar la “matriz cuadrada”

Los números primos se ven reflejados en las esquinas, saliendo del 7, 5, 3. Donde se traza una línea con pendiente m, para encontrar los demás números primos.

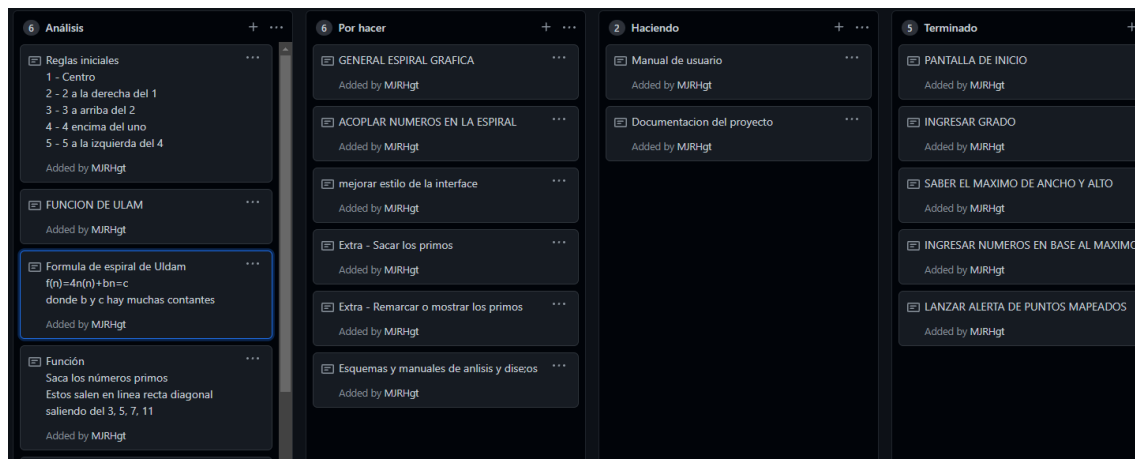
En dicho proyecto se plantea que todo esto sea, a través de una petición de numero al usuario, donde nuestro caso base es el número 100.

La fórmula matemática de la espiral de ulam, es $f(x)=4n^2 + bn$, donde b es constante

Metodología de trabajo

Se utilizo github, para subir el proyecto, pues podríamos subir sin mayor problema, el .exe, el .asm y el .obj. a su vez los archivos tasm.exe y tlink.exe y la documentación en formato PDF

Donde también generamos un tablero, para llevar control de los procesos realizados, dicho tablero, quedo con tareas pendientes, que es para mejorar en la parte grafica del proyecto, verificaciones de calidad, pero que ayudo con el análisis y no intervenir en tareas de otros.



Código

El efecto práctico del código es usar x,y, para generar los parámetros del plano cartesiano, aprender a validar datos de inspección del número, a su vez cómo funciona la interrupción 10h y cómo limpiar el registro BH.

En la siguiente imagen se evidencia cómo se lee cada cantidad de numérica dentro de tasm

```
;-----leer cantidad de puntos-----
reading:
    ;centenas
    mov bl, 64h                ;*100
    call readDigit
    mov points, ax
    ;decenas
    mov bl, 0ah                ;*10
    call readDigit
    add points, ax
    ;unidades
    mov bl, 01h                ;*1
    call readDigit
    add points, ax
    call printNum
```

Y en la siguiente imagen se ve cómo se evalúa el criterio de que número es aceptado, teniendo que llamar a la validación del error, por si no cumple

```
;evaluar rango
mov ax, points
cmp ax, 64h
jg outRange                ;si el points - 100 >= 0
cmp ax, 0h
jne inRange                ;si el points - 100 <= 0
outRange:
    lea dx, errorString
    call printString
    call printNum
    lea dx, instruction
    call printString
    jmp reading
inRange:
```

Y en esta ultima imagen podemos ver los movimientos de cada uno de los números en su espacio del plano cartesiano

```
cmp points, 01h
je spiral
mov bh, 01h
mov cx, 0h
ciclowhile:
    cmp direction, 0h                ;direction = 0? (derecha)
    je aumentarmovs
    cmp direction, 2h                ;direction = 2? (izquierda)
    je aumentarmovs
    jne setcx
    aumentarmovs:                    ;aumentar movs si va para derecha o izquierda
    inc movements
    setcx:
    mov cl, movements
    ciclofor:
        cmp direction, 0h
        je right
        cmp direction, 01h
        je up
        cmp direction, 02h
        je left
        ;si no es ninguna de las anteriores, es abajo
        dec positionY
        jmp continuar
    right:
        inc positionX
        jmp continuar
    up:
        inc positionY
        jmp continuar
    left:
        dec positionX
        jmp continuar
    continuar:
    inc bh
    ;imprimir punto y coordenada
    call imprimirptocoordenada
```

Posterior a eso, se imprime cada numero con su coordenada por medio de interrupción 21h y los distintos parámetros, de modo que en consola se vea algo parecido a esto

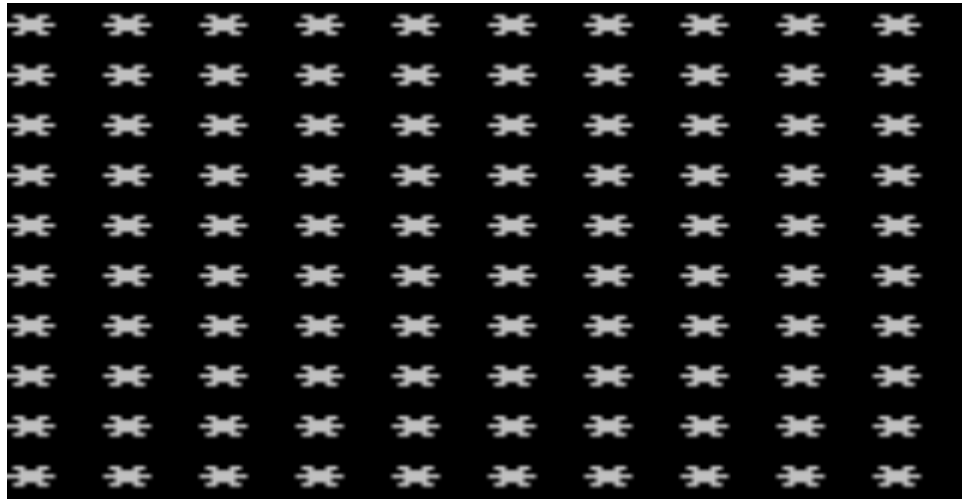
“coordenada 001 <0,0>”

Imágenes del programa

1. Ingreso del numero

```
C:\Documents and Settings\Administrator\Desktop>project.exe  
Ingrese un numero  
100
```

2. Generar puntos del plano



3. Generar coordenadas

```
Coordenada 001 <0,0>  
Coordenada 002 <1,0>  
Coordenada 003 <1,1>  
Coordenada 004 <0,1>  
Coordenada 005 <-1,1>  
Coordenada 006 <-1,0>  
Coordenada 007 <-1,-1>
```


Llegando hasta el final, del número ingresado

```
Coordenada 091 <5,5>
Coordenada 092 <4,5>
Coordenada 093 <3,5>
Coordenada 094 <2,5>
Coordenada 095 <1,5>
Coordenada 096 <0,5>
Coordenada 097 <-1,5>
Coordenada 098 <-2,5>
Coordenada 099 <-3,5>
Coordenada 100 <-4,5>

C:\DOCUME~1\ADMINI~1\Desktop>
```

Puntos de mejora

1. Interfaz gráfica, en cara al usuario
2. Insertar los números dentro de la espiral
3. Mejorar interfaz de la espiral
4. Remarcar los números primos

Conclusión

Dicho proyecto es retador en muchos sentidos, sobre todo en el manejo numérico y correcta implementación de los saltos y ciclos. Se logro utilizar interrupciones de memoria y lograr validad números válidos.

Donde podemos resaltar que se logro el cometido es en encontrar las coordenadas de cada número, generando de manera visual para el usuario una guía de gráfica, donde quedamos a deber fue en esto último, en la parte grafica del Proyecto.