Ejercicio 1.1.- Mínimo de 3 valores

```
def run(value1: int | float, value2: int | float, value3: int | float) -> int | float:
    if value1 < value2 and value1 < value3:
        min_value = value1
    elif value2 < value1 and value2 < value3:
        min_value = value2
    else:
        min_value = value3
    return min_value

resultado = run(value1=100,value2=50.5,value3=30.3)
print("minimo", resultado )
```

```
minimo 30.3
```

Ejercicio 1.2.- Donación de sangre

```
def run(age: int, weight: int, heartbeat: int, platelets: int) -> bool:
    suitable_for_donation = True
    if age < 18 or age > 65:
        suitable_for_donation = False
        print("Eres demasiado joven o muy viejo")
    if weight < 50:
        suitable_for_donation = False
        print("Peso demasiado bajo")
    if heartbeat < 50 or heartbeat > 110:
        suitable_for_donation = False
        print("Pulso anormal")
    if platelets < 125000:
        suitable_for_donation = False
        print("No tienes suficientes plaquetas")
    if platelets < 135000 and platelets > 125000:
        print("Puedes donar si eres mujer")
    return suitable_for_donation

resultado = run(age=45, weight=70, heartbeat=120, platelets=410000)
print("¿Apto para donar sangre?", resultado)
```

```
Pulso anormal
¿Apto para donar sangre? False
```

Ejercicio 1.3.- Atajos de teclado

```
def run(key1: str, key2: str, key3: str) -> str:
    keys = []
    if key1 != "":
        keys.append(key1)
    if key2 != "":
        keys.append(key2)
    if key3 != "":
        keys.append(key3)
    action = "+".join(keys)
    actions = {
         "CTRL+ALT+T": "Open terminal",
         "CTRL+ALT+L": "Lock screen",
         "CTRL+ALT+D": "Show desktop",
         "ALT+F2": "Run console",
         "CTRL+Q": "Close window",
         "CTRL+ALT+DEL": "Log out"
    }
    if action in actions:
        print(actions[action])
    else:
        print("Undefined")
    return action
resultad = run(key1="CTRL", key2="ALT", key3="D")
print("Atajo",resultad )
```

```
Show desktop
Atajo CTRL+ALT+D
```

```
def run(key1: str, key2: str, key3: str) -> str:
    keys = []
    if key1 != "":
        keys.append(key1)
    if key2 != "":
```

```
        keys.append(key2)
    if key3 != "":
        keys.append(key3)
    action = "+".join(keys)

    if action == "CTRL+ALT+T":
        return "Open terminal"
    elif action == "CTRL+ALT+L":
        return "Lock screen"
    elif action == "CTRL+ALT+D":
        return "Show desktop"
    elif action == "ALT+F2":
        return "Run console"
    elif action == "CTRL+Q":
        return "Close window"
    elif action == "CTRL+ALT+DEL":
        return "Log out"
    else:
        return "Undefined"
    return action
resultado = run(key1="CTRL", key2="", key3="")
print(resultado )
```

```
Undefined
```

Ejercicio 1.4.- Sumando múltiplos

```
def run(limit: int) -> None:
  multiplos=[]
  suma=0
  for i in range(0,limit,3):
    multiplos.append(i)
    suma += i
    if suma<limit:
      print(suma)
    else:
      return
  print(suma)

resultado = run(limit=100)
print(resultado )
```

```
0
3
9
18
30
45
63
84
None
```

```
limit = 70
suma = 0
multiplos = ""

for i in range(0, limit, 3):
      multiplos += str(i) + " "
      suma += i
      if suma >= limit:
          break
print(multiplos)
print(suma)
```

```
0 3 6 9 12 15 18 21
84
```

Ejercicio 1.5.- El árbol del 1

```
n = 4
for i in range(1, n + 1):
    ascendente = ""
    for j in range(1, i + 1):
        ascendente += str(j)
    descendente = ""
    for j in range(i - 1, 0, -1):
        descendente += str(j)
    print(ascendente+descendente)
```

```
1
121
12321
1234321
```

```
n = 9
filas = []
for i in range(1, n + 1):
    ascendente = ""
    for j in range(1, i + 1):
        ascendente += str(j)
    descendente = ""
    for j in range(i - 1, 0, -1):
        descendente += str(j)
    filas.append(ascendente + descendente)
print('\n'.join(filas))
```

```
1
121
12321
1234321
123454321
12345654321
1234567654321
123456787654321
12345678987654321
```

Ejercicio 1.6.- Caballo de ajedrez

```
def run(target_x: int, target_y: int) -> int:
  pos_x = 5
  pos_y = 4
  movements = 0
  while True:
    if pos_x == target_x and pos_y == target_y:
      print(f"Resuelto en {movements} movimientos")
      return movements
    elif pos_x > target_x or pos_y > target_y:
      print("No se puede resolver")
      return -1
    if target_x - pos_x > target_y - pos_y:
        pos_x = pos_x + 2
        pos_y = pos_y + 1
    else:
        pos_x = pos_x + 1
        pos_y = pos_y + 2
    movements += 1
    print(f"Posición: ({pos_x}, {pos_y}), Movimientos: {movements}")

resultado = run(0, 0)
print("Resultado:", resultado)
```

```
No se puede resolver
Resultado: -1
```

Ejercicio 1.7.- Adivina un número

```
def run(target_number: int) -> None:
  intentos = 0

  while True:
      user_input = input("Introduzca número:")
      numero = int(user_input)
      intentos += 1
      if numero == target_number:
          print(f"Enhorabuena has encontrado el número en {intentos} intentos\n")
          break
      elif numero < target_number:
          print("Mayor")
      else:
          print("Menor")
run(87)
```

```
Introduzca número:47
Mayor
Introduzca número:87
Enhorabuena has encontrado el número en 2 intentos
```

Ejercicio 1.8.- Distancia de Hammming

```
def run(text1: str, text2: str) -> int:
  dhamming = 0
  if len(text1) != len(text2):
      return -1
  else:
      for i in range(len(text1)):
          if text1[i] != text2[i]:
              dhamming += 1
      return dhamming
run('0001010011101', '0000110010001')
```

```
4
```

Ejercicio 2.- Diagrama de flujo de la sucesión de Fibonacci

```
def run(n: int) -> float:
  fibo=0
  a = 1
  b = 1
  for i in range(n):
      fibo = a
      c = a + b
      a = b
      b = c
      print(fibo)
run(10)
```

```
1
1
2
3
5
8
13
21
34
55
```

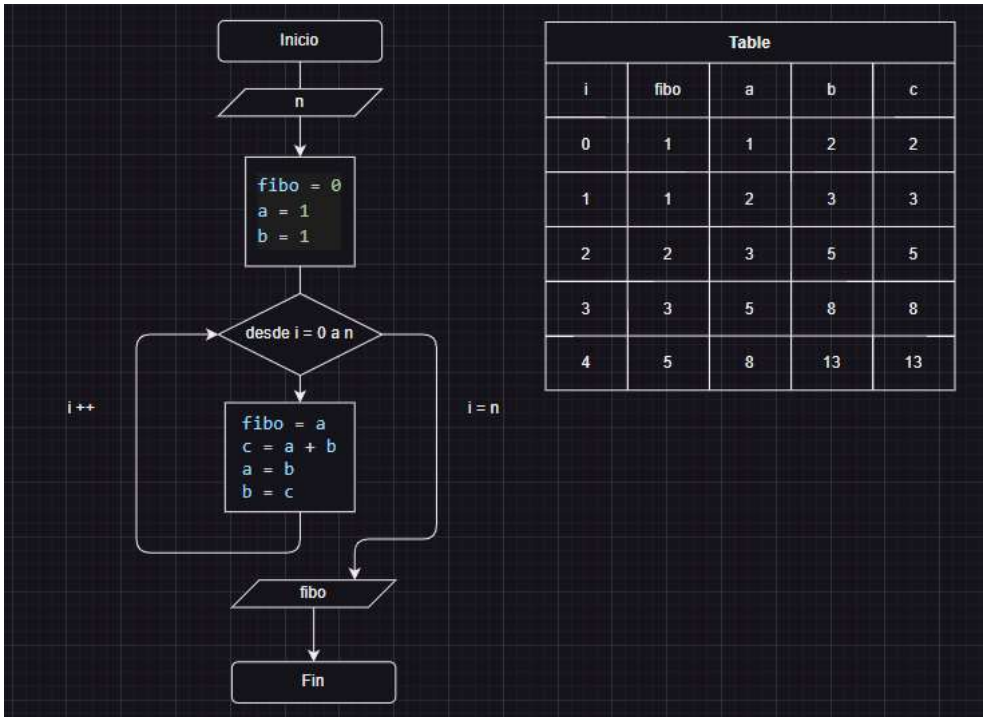


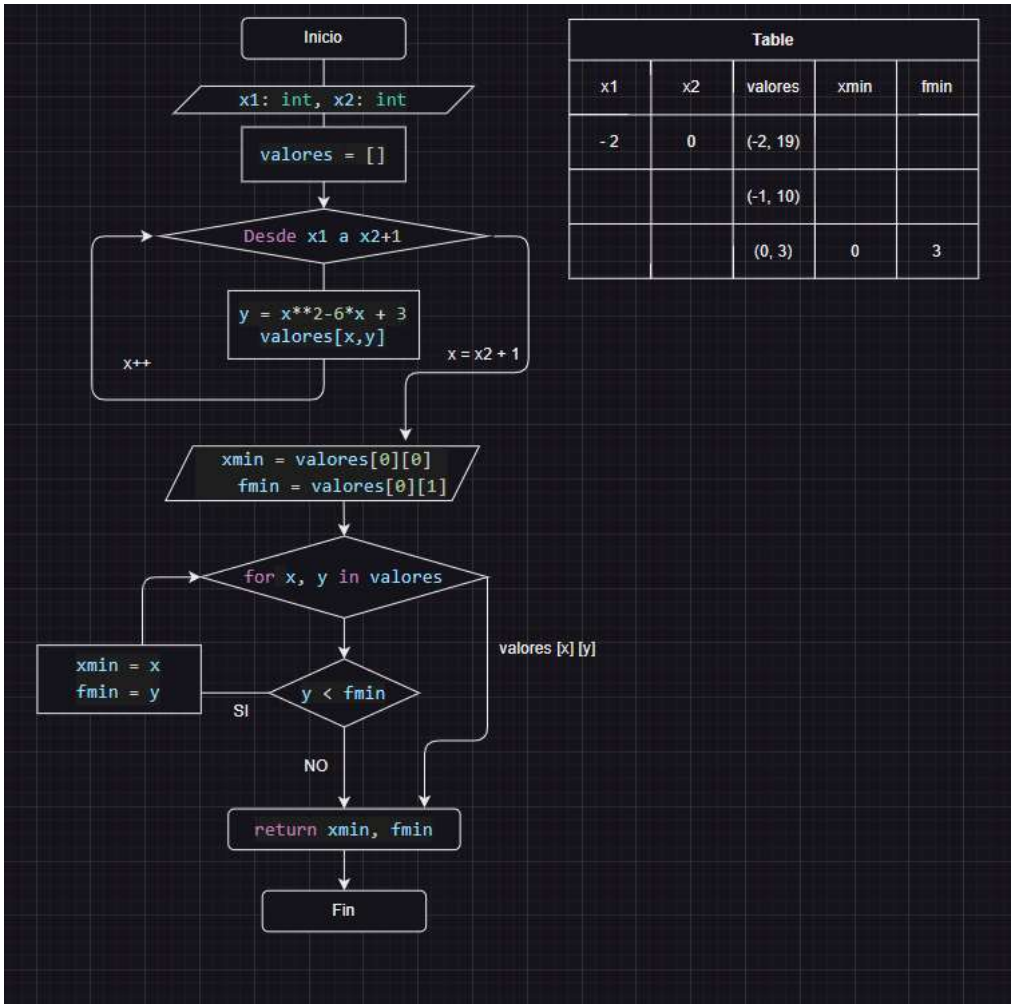
Table				
i	fibo	a	b	c
0	1	1	2	2
1	1	2	3	3
2	2	3	5	5
3	3	5	8	8
4	5	8	13	13

Ejercicio 3.- Diagrama de flujo del Mínimo de una función

```
def run(x1: int, x2: int) -> tuple:
  valores = []
  for x in range(x1, x2+1):
    y = x**2-6*x +3
    valores.append((x, y))
```

```
    print(valores)
    xmin = valores[0][0]
    fmin = valores[0][1]
    for x, y in valores:
      if y < fmin:
        xmin= x
        fmin= y
    return xmin, fmin
run(-9, 9)
```

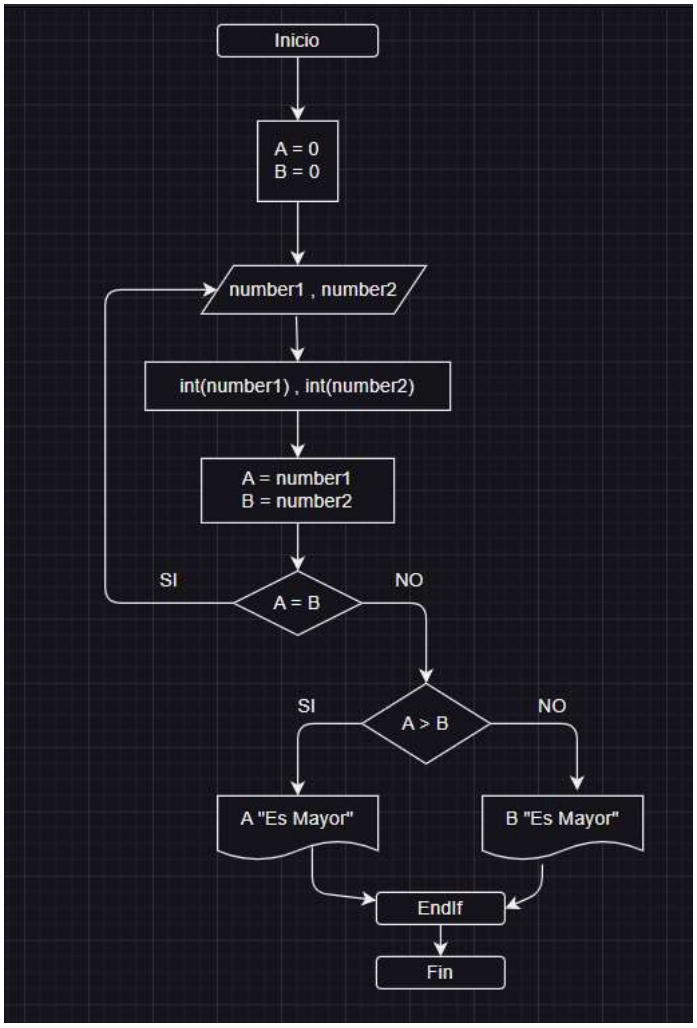
```
[(-9, 138), (-8, 115), (-7, 94), (-6, 75), (-5, 58), (-4, 43), (-3, 30), (-2, 19), (-1, 10), (0, 3), (1, -2), (2, -5), (3, -6), (4,
(3, -6)
```



Table

x1	x2	valores	xmin	fmin
-2	0	(-2, 19)		
		(-1, 10)		
		(0, 3)	0	3

Ejercicio 4.- Diagrama de flujo en base a un pseudocódigo


Inicio
A = 0
B = 0
number1 , number2
int(number1) , int(number2)
A = number1
B = number2
SI
A = B
NO
SI
A > B
NO
A "Es Mayor"
B "Es Mayor"
EndIf
Fin