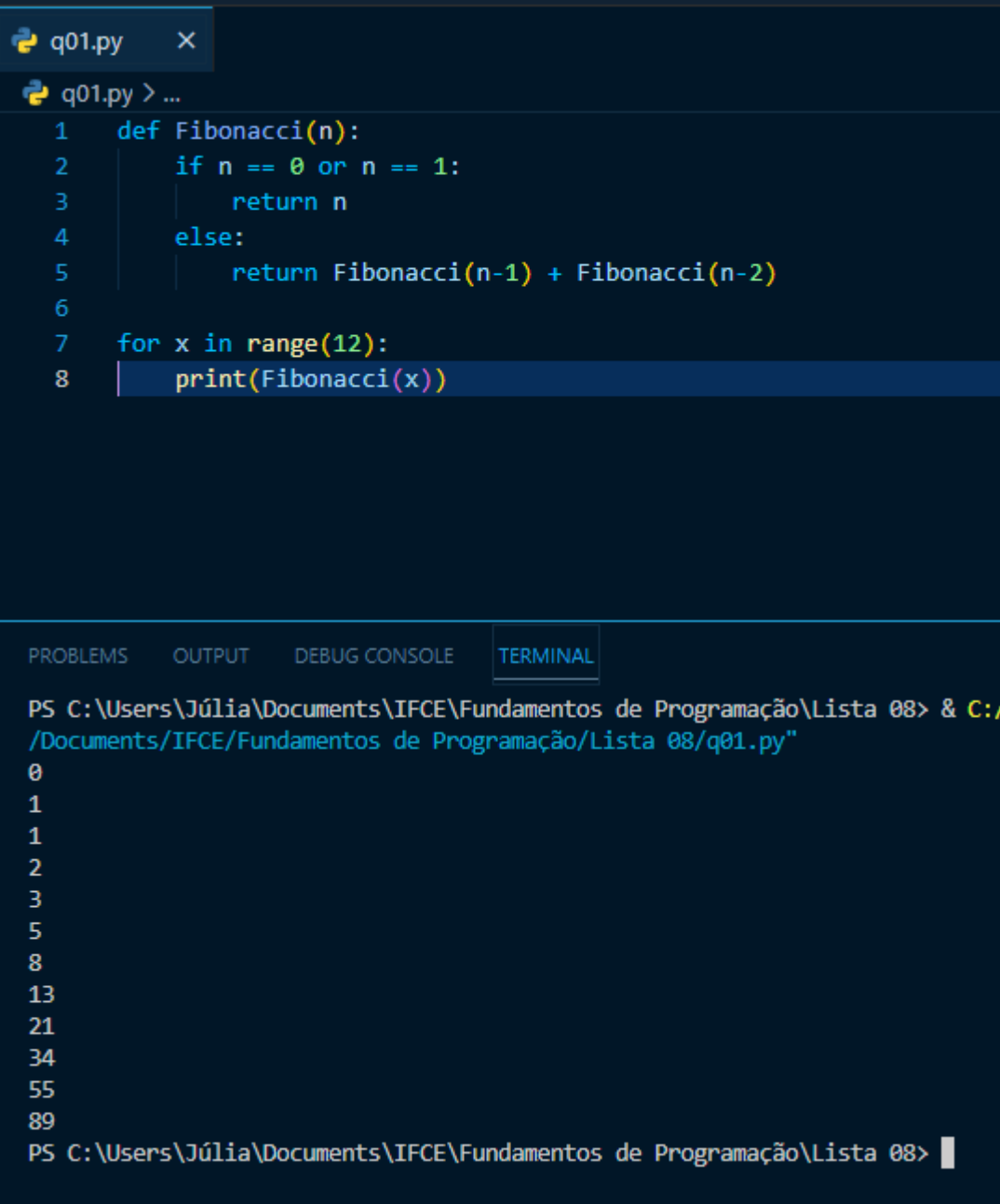


01)



The image shows a Python IDE window with a file named 'q01.py'. The code defines a 'Fibonacci' function and a loop to print the first 12 Fibonacci numbers. The output in the terminal shows the sequence: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89.

```
q01.py X
q01.py > ...
1 def Fibonacci(n):
2     if n == 0 or n == 1:
3         return n
4     else:
5         return Fibonacci(n-1) + Fibonacci(n-2)
6
7 for x in range(12):
8     print(Fibonacci(x))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\Júlia\Documents\IFCE\Fundamentos de Programação\Lista 08> & C:/Python/Python39/python.exe C:/Users/Júlia/Documents/IFCE/Fundamentos de Programação/Lista 08/q01.py"
0
1
1
2
3
5
8
13
21
34
55
89
PS C:\Users\Júlia\Documents\IFCE\Fundamentos de Programação\Lista 08> |
```

02)

```
q02.py ×
q02.py > ...
1 def SomaDiagonal(matriz):
2     sum = 0
3     for i in range(3):
4         for p in range(3):
5             print(f'{matriz[i][p]:^5}', end=" ")
6             print("")
7
8     for i in range(0, 3):
9         print(matriz[i][i])
10        sum += (matriz[i][i])
11    print(f'A soma da diagonal principal é: {sum}')
12
13
14
15 matriz = [[9, 4, 7], [5, 1, 8], [0, 3, 9]]
16 SomaDiagonal(matriz)
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL**

```
PS C:\Users\Júlia\Documents\IFCE\Fundamentos de Programação\Lista 08> & C:/User
/Documents/IFCE/Fundamentos de Programação/Lista 08/q02.py"
  9    4    7
  5    1    8
  0    3    9
  9
  1
  9
A soma da diagonal principal é: 19
PS C:\Users\Júlia\Documents\IFCE\Fundamentos de Programação\Lista 08> █
```

03)

```
q03.py x
q03.py > ...
1  from random import randint
2
3  def GeraMatriz(dimensao):
4      matriz = []
5      for i in range(0, dimensao):
6          matriz.append([randint(0,100), randint(1,100), randint(0,100)])
7      return matriz
8
9  def Vetor(matriz):
10     vetor = []
11     for i in range(0, len(matriz)):
12         for j in range(0, len(matriz)):
13             vetor.append(matriz[i][j])
14     return vetor
15
16 def MaiorValor(vetor, tamanho, i):
17     if tamanho == 0:
18         return vetor[i]
19     else:
20         if vetor[tamanho-1] > vetor[i]:
21             return MaiorValor(vetor, tamanho-1, tamanho-1)
22         else:
23             return MaiorValor(vetor, tamanho-1, i)
24
25
26 matriz = GeraMatriz(3)
27 print(matriz)
28
29 vetor = Vetor(matriz)
30 tvetor = len(vetor)
31 print(MaiorValor(vetor, tvetor, 0))
32
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\Júlia\Documents\IFCE\Fundamentos de Programação\Lista 08> & C:/U
/Documents/IFCE/Fundamentos de Programação/Lista 08/q03.py"
[[17, 95, 49], [13, 75, 59], [37, 68, 34]]
95
PS C:\Users\Júlia\Documents\IFCE\Fundamentos de Programação\Lista 08> |
```

04)

q04.py X

q04.py > exponencial

```
1 def exponencial(base, exp):
2     if exp == 0:
3         return 1
4
5     else:
6         return base * exponencial(base, exp-1)
7
8
9 print(exponencial(2, 0))
10 print(exponencial(2, 4))
11 print(exponencial(2, 9))
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

/Documents/IFCE/Fundamentos de Programação/Lista 08/q04.py"

1

16

512

PS C:\Users\Júlia\Documents\IFCE\Fundamentos de Programação\Lista 08> [