

This screenshot shows the Visual Studio Code editor with a Python file named 7A.py. The code defines a factorial function and a driver program to test it. The function checks if the input is 1 or 0, returns 1, or recursively calculates the factorial. The driver program sets a number to 5 and prints the result.

```
1 def factorial(n):
2
3     # Checking the number # is 1 or 0 then
4     # return 1
5     # other wise return # factorial
6     if (n==1 or n==0):
7
8         return 1
9
10    else:
11
12        return (n * factorial(n - 1))
13
14 # Driver Code
15 num = 5;
```

The terminal output shows the execution of the program, resulting in the factorial of 5 being 120.

```
PS C:\Users\acer\Desktop\python> python 7A.py
number : 5
Factorial : 120
PS C:\Users\acer\Desktop\python>
```

This screenshot shows the Visual Studio Code editor with a Python file named 8A.py. The code defines a function to find the maximum element in a list and a driver program to test it. The function uses a loop to iterate through the list and find the maximum value. The driver program sets a list and prints the maximum value.

```
1 def find_max(l):
2     # end of function lcs
3     for j in range(n + 1):
4         if i == 0 or j == 0 :
5             l[i][j] = 0
6         elif x[i-1] == y[j-1]:
7             l[i][j] = l[i-1][j-1]+1
8         else:
9             l[i][j] = max(l[i-1][j], l[i][j-1])
10    return l[i][j]
```

The terminal output shows the execution of the program, resulting in the maximum value of the list being 3000.

```
PS C:\Users\acer\Desktop\python> python 8A.py
length of lcs is 4
PS C:\Users\acer\Desktop\python>
```

This screenshot shows the Visual Studio Code editor with a Python file named 8A.py. The code defines a function to generate the Fibonacci sequence and a driver program to test it. The function checks if the number of terms is valid and generates the sequence. The driver program sets the number of terms to 10 and prints the sequence.

```
1 nterms = int(input("How many terms? "))
2 # first two terms
3 n1, n2 = 0, 1
4 count = 0
5
6 # check if the number of terms is valid
7 if nterms <= 0:
8     print("Please enter a positive integer")
9 # if there is only one term, return n1
10 elif nterms == 1:
11     print("Fibonacci sequence upto",nterms,":")
12     print(n1)
13 # generate fibonacci sequence
14 else:
15     print("Fibonacci sequence:")
```

The terminal output shows the execution of the program, resulting in the Fibonacci sequence up to 10 terms being printed.

```
PS C:\Users\acer\Desktop\python> python 8A.py
How many terms? 10
Fibonacci sequence:
0
1
1
2
3
5
8
13
21
34
PS C:\Users\acer\Desktop\python>
```

```
>>> =====
>>>
Minimum element is 1
Maximum element is 3000
>>>
```