

# LAB ASSIGNMENT-3.4

## TASK-1:

### PROMPT:

Generate a python function to print first N Fibonacci numbers

### CODE:

```
def fibonacci(n):
    a, b = 0, 1
    for i in range(n):
        print(a, end=" ")
        a, b = b, a + b
```

```
fibonacci(7)
```

### OUTPUT:

The screenshot shows a code editor with a dark theme. The script file `LAB-3.4.py` contains the following code:

```
LAB-3.4.py > ...
1 #Generate a python function to print first N Fibonacci numbers
2 def fibonacci(n):
3     a, b = 0, 1
4     for i in range(n):
5         print(a, end=" ")
6         a, b = b, a + b
7
8 fibonacci(7)
9
10
11
12
```

Below the code editor is a terminal window showing the execution of the script and its output:

```
PROBLEMS 20 OUTPUT DEBUG CONSOLE TERMINAL GITLENS SPELL CHECKER 14
● PS C:\Users\madhu\Desktop\Ai assistant> & C:/Users/madhu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/madhu/Desktop/Ai assistant\LAB-3.4.py"
0 1 1 2 3 5 8
● PS C:\Users\madhu\Desktop\Ai assistant> & C:/Users/madhu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/madhu/Desktop/Ai assistant\LAB-3.4.py"
0 1 1 2 3 5 8
○ PS C:\Users\madhu\Desktop\Ai assistant>
```

### EXPLANATION:

- The function generates Fibonacci numbers using two variables.
- A loop runs N times and prints numbers one by one.

## TASK-2

### PROMPT:

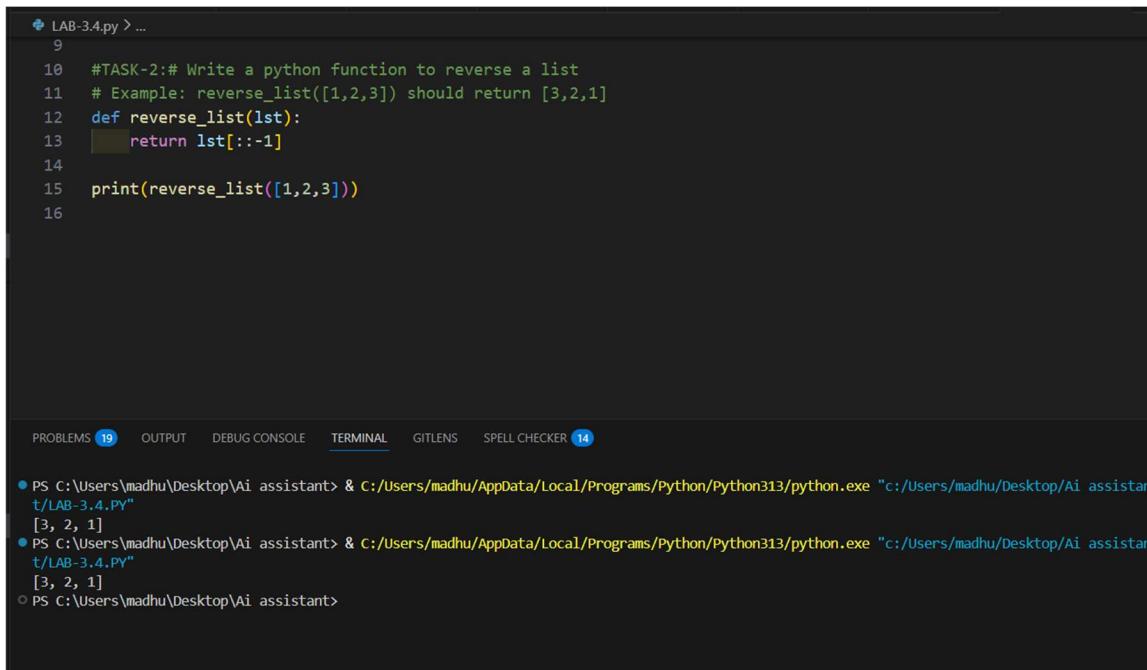
```
# Write a python function to reverse a list
```

```
# Example: reverse_list([1,2,3]) should return [3,2,1]
```

#### CODE:

```
def reverse_list(lst):  
    return lst[::-1]  
  
print(reverse_list([1,2,3]))
```

#### OUTPUT:



The screenshot shows a code editor interface with a dark theme. At the top, there's a file tab labeled "LAB-3.4.py > ...". Below the editor area, there are tabs for "PROBLEMS" (19), "OUTPUT", "DEBUG CONSOLE", "TERMINAL" (selected), "GITLENS", and "SPELL CHECKER" (14). In the terminal pane, the following command and output are shown:

```
PS C:\Users\madhu\Desktop\Ai assistant> & c:/Users/madhu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/madhu/Desktop/Ai assistant/LAB-3.4.py"  
[3, 2, 1]  
PS C:\Users\madhu\Desktop\Ai assistant> & c:/Users/madhu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/madhu/Desktop/Ai assistant/LAB-3.4.py"  
[3, 2, 1]  
PS C:\Users\madhu\Desktop\Ai assistant>
```

#### EXPLANATION:

- The function reverses the given list.
- It uses slicing method to change order of elements.
- The output list is returned in reverse format.

### TASK-3:

#### PROMPT:

Write a function `is_valid()` that returns True if the string starts with a capital letter and ends with a period. Ask the user about their past behavior (this is their history)

#### CODE:

```
def is_valid(s)  
    return s[0].isupper() and s.endswith('.')
```

```
print(is_valid("Hello."))

print(is_valid("hello."))

print(is_valid("Hello"))
```

## OUTPUT:

The screenshot shows a code editor interface with a dark theme. On the left is a code editor pane displaying a Python script named 'LAB-3.4.py'. The code defines a function 'is\_valid' that checks if a string starts with a capital letter and ends with a period. It then prints the result of calling this function with three different strings: "Hello.", "hello.", and "Hello". Below the code editor is a terminal pane showing the execution of the script and its output. The terminal tab is active, indicated by a blue underline. The output shows two runs of the script. In the first run, it prints 'True' for "Hello." and 'False' for both "hello." and "Hello". In the second run, it prints 'True' for "Hello." and 'False' for both "hello." and "Hello". The terminal also shows the command 'python.exe' being used to run the script.

```
LAB-3.4.py > ...
16
17 #TASK-3: Write a function is_valid() that returns True and The string starts with a capital letter and ends with a
18 def is_valid(s):
19     return s[0].isupper() and s.endswith('.')
20
21 print(is_valid("Hello."))
22 print(is_valid("hello."))
23 print(is_valid("Hello"))
24

PROBLEMS 19 OUTPUT DEBUG CONSOLE TERMINAL GITLENS SPELL CHECKER 14

● PS C:\Users\madhu\Desktop\Ai assistant> & C:/Users/madhu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/madhu/Desktop/Ai assistant/LAB-3.4.PY"
True
False
False
● PS C:\Users\madhu\Desktop\Ai assistant> & C:/Users/madhu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/madhu/Desktop/Ai assistant/LAB-3.4.PY"
True
False
False
○ PS C:\Users\madhu\Desktop\Ai assistant>
```

## EXPLANATION:

- The function checks whether the first letter is capital.
- It also checks whether the string ends with a period.
- Both conditions must be true to return True.

**TASK-4:****PROMPT:**

Write a python function to validate an email address

**CODE:**

```
import re
```

```
def validate_email(email):
    pattern = r"^[a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+$"
    if re.match(pattern, email):
        return True
    else:
        return False
```

```
email = input("Enter email address: ")
```

```
if validate_email(email):
    print("Valid Email")
else:
    print("Invalid Email")
```

## OUTPUT:

The screenshot shows a code editor with a dark theme. The code in the editor is as follows:

```
LAB-3.4.py > validate_email
24
25 # TASK-4:Write a python function to validate an email address
26 import re
27
28 def validate_email(email):
29     pattern = r"^[a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+$"
30     if re.match(pattern, email):
31         return True
32     else:
33         return False
34
35
36 email = input("Enter email address: ")
37
38 if validate_email(email):
39     print("Valid Email")
40
```

Below the code, there is a terminal window showing the execution of the script and its output:

- PS C:\Users\madhu\Desktop\Ai assistant> & C:/Users/madhu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/madhu/Desktop/Ai assistant/LAB-3.4.PY"
- Enter email address: Traceback (most recent call last):
  - File "c:/Users/madhu/Desktop/Ai assistant/LAB-3.4.PY", line 36, in <module>
  - email = input("Enter email address: ")  
KeyboardInterrupt
- PS C:\Users\madhu\Desktop\Ai assistant> & C:/Users/madhu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/madhu/Desktop/Ai assistant/LAB-3.4.PY"
- Enter email address: madhuri@11  
Invalid Email
- PS C:\Users\madhu\Desktop\Ai assistant>

## EXPLANATION:

- The function checks if email contains “@” symbol.
- It also checks if email contains a dot (.).
- This is a simple and basic validation method.

## TASK-5:

### PROMPT:

Write a function to return sum of digits of a number

### CODE:

```
def sum_of_digits(n):
```

```
    total = 0
```

```
    while n > 0:
```

```
        total += n % 10
```

```
        n = n // 10
```

```
    return total
```

```
print(sum_of_digits(123))
```

## OUTPUT:

```
43  # TASK-5:Write a function to return sum of digits of a number
44  def sum_of_digits(n):
45      total = 0
46      while n > 0:
47          total += n % 10
48          n = n // 10
49  return total
50
51 print(sum_of_digits(123))
52
53
54
```

PROBLEMS 19 OUTPUT DEBUG CONSOLE TERMINAL GITLENS SPELL CHECKER 14

```
PS C:\Users\madhu\Desktop\Ai assistant> & C:/Users/madhu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/madhu/Desktop/Ai assistant/LAB-3.4.py"
6
PS C:\Users\madhu\Desktop\Ai assistant> & C:/Users/madhu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/madhu/Desktop/Ai assistant/LAB-3.4.py"
6
```

### EXPLANATION:

- The function extracts digits using loop.
- Each digit is added to total sum.
- Finally the sum of digits is returned.