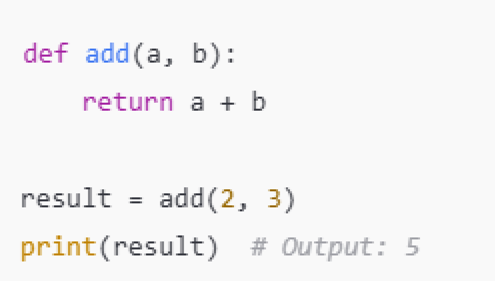
1. **What is the difference between a function and a method in Python?**
   A function is a block of code that performs a specific task.
   A method is a function that is associated with an object.
2. **Explain the concept of function arguments and parameters in Python.**

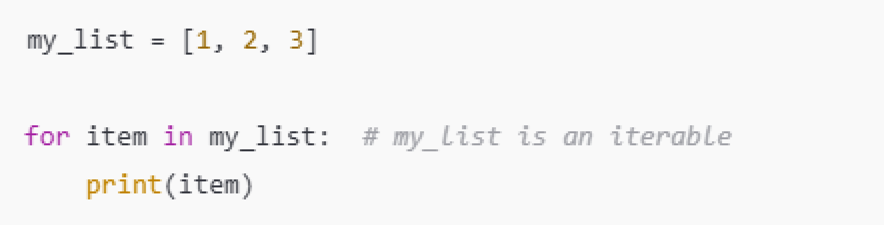
Parameters are the names used in a function definition to specify what kind of arguments the function can accept. Arguments are the actual values you pass into a function when calling it.

3. **What are the different ways to define and call a function in Python?**
   we can standardize a function by using **def.**
   we can use anonymous functions by using **lamda**
   we can use positional arguments like calling functions.
4. **What is the purpose of the `return` statement in a Python function?**
   It gives back the result of the function for further use:



```
def add(a, b):
    return a + b

result = add(2, 3)
print(result)  # Output: 5
```

5. **What are iterators in Python and how do they differ from iterables?**
   An **iterable** is any object that **can be looped over** (used in a for loop).



```
my_list = [1, 2, 3]

for item in my_list:  # my_list is an iterable
    print(item)
```

6. **Explain the concept of generators in Python and how they are defined.**
   A **generator** is a special type of function that **produces values on the fly** using the yield keyword instead of return.

- Generators **do not store** all values in memory.
- They are **lazy**, meaning they only compute values when requested.

7. **What are the advantages of using generators over regular functions?**

Generators yield items one at a time instead of storing the entire sequence in memory like lists or regular functions that return collections.

8. **What is a lambda function in Python and when is it typically used?**

   A **lambda function** in Python is a **small anonymous function** defined using the `lambda` keyword instead of `def`. It can have **any number of arguments** but only **one expression**, which is implicitly returned.

9. **Explain the purpose and usage of the `map()` function in Python**

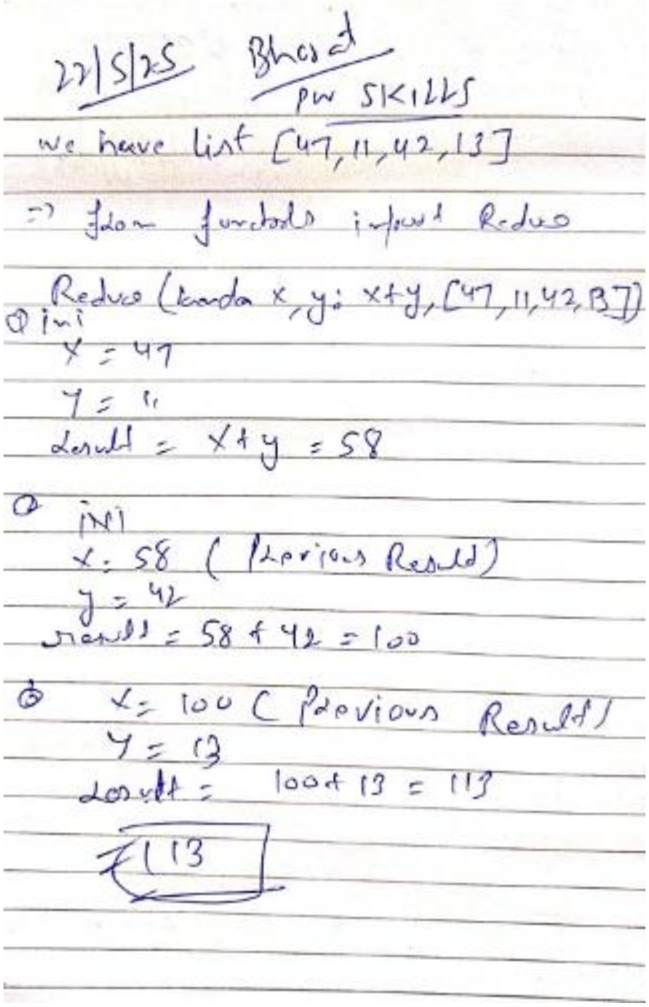
   The map() function is a **built-in Python function** used to **apply a function to every item in an iterable** (like a list, tuple, etc.) and return a **map object** (which is an iterator).

10. **What is the difference between `map()`, `reduce()`, and `filter()` functions in Python?**

Function	Purpose	Output
map()	Transforms each item in an iterable	New iterable with transformed items
filter()	Selects items based on a condition	New iterable with filtered items
reduce()	Aggregates items to a single result	A single cumulative value

11. **Using pen & Paper write the internal mechanism for sum operation using reduce function on this given list:[47,11,42,13];**

Image given below



22/5/25 Bharat
pw SKILLS

we have list [47, 11, 42, 13]

=> from functools import Reduce

Reduce (lambda x, y: x+y, [47, 11, 42, 13])

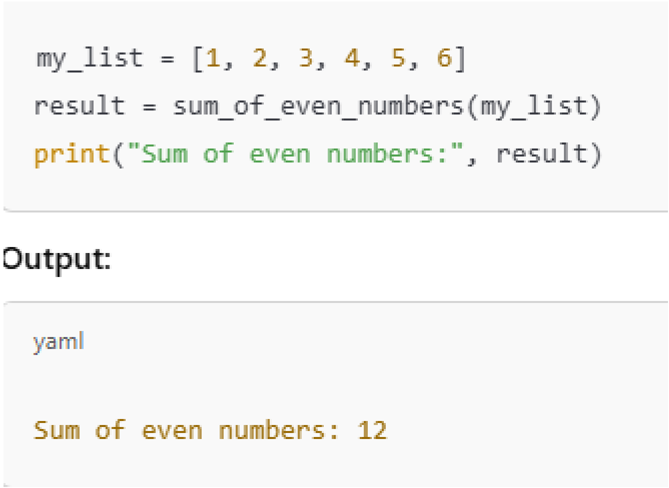
① ini
x = 47
y = 11
Result = x+y = 58

② ini
x: 58 (Previous Result)
y = 42
result = 58 + 42 = 100

③ x = 100 (Previous Result)
y = 13
Result = 100+13 = 113

113

1. Write a Python function that takes a list of numbers as input and returns the sum of all even numbers in the list.



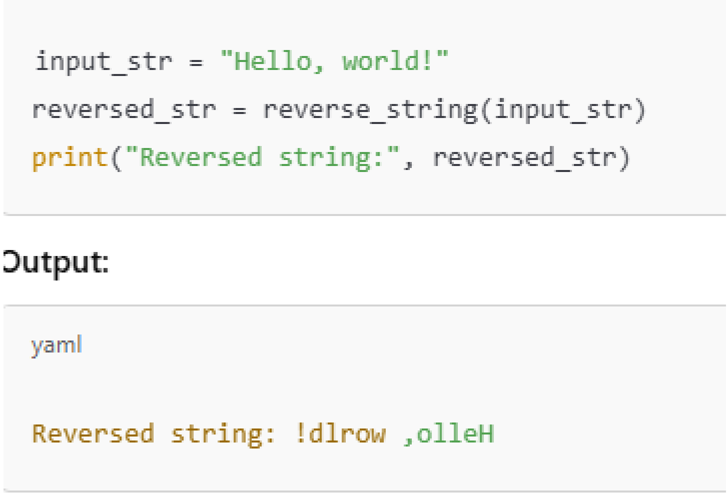
```
my_list = [1, 2, 3, 4, 5, 6]
result = sum_of_even_numbers(my_list)
print("Sum of even numbers:", result)
```

Output:

yaml

```
Sum of even numbers: 12
```

2. Create a Python function that accepts a string and returns the reverse of that string.

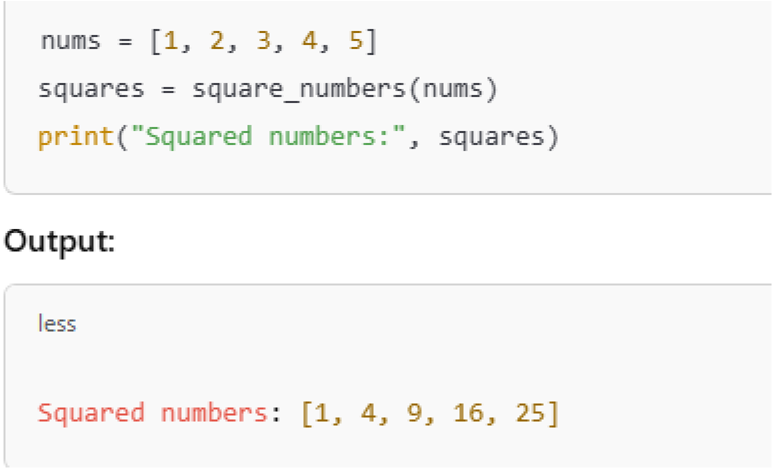


```
input_str = "Hello, world!"
reversed_str = reverse_string(input_str)
print("Reversed string:", reversed_str)
```

Output:

```
Reversed string: !dlrow ,olleH
```

3. Implement a Python function that takes a list of integers and returns a new list containing the squares of each number.

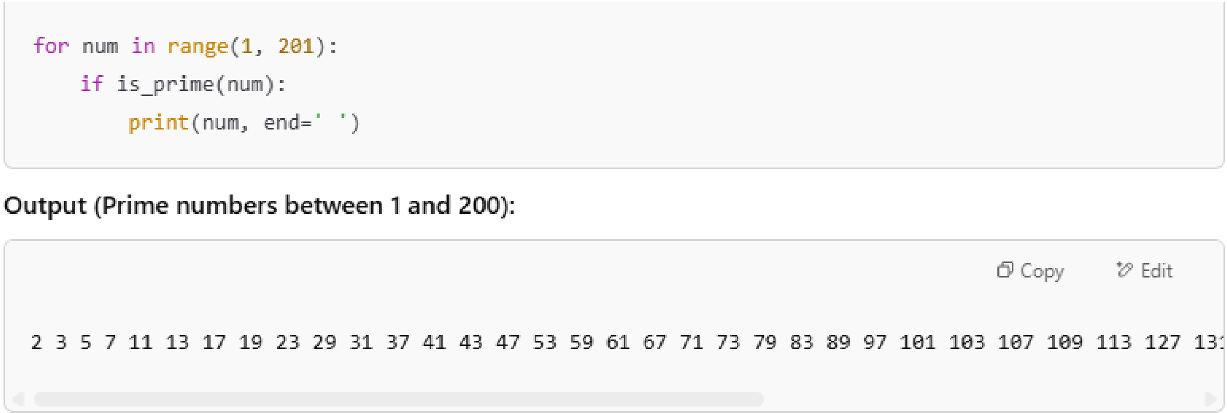


```
nums = [1, 2, 3, 4, 5]
squares = square_numbers(nums)
print("Squared numbers:", squares)
```

Output:

```
Squared numbers: [1, 4, 9, 16, 25]
```

4. Write a Python function that checks if a given number is prime or not from 1 to 200.

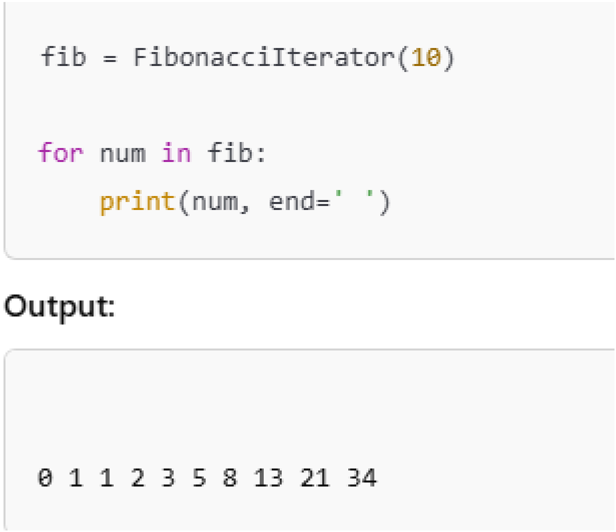


```
for num in range(1, 201):
    if is_prime(num):
        print(num, end=' ')
```

Output (Prime numbers between 1 and 200):

```
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 101 103 107 109 113 127 13
```

5. Create an iterator class in Python that generates the Fibonacci sequence up to a specified number of terms.

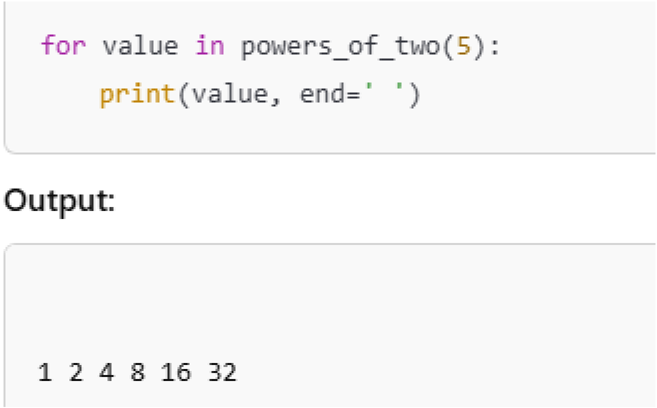


```python
fib = FibonacciIterator(10)

for num in fib:
    print(num, end=' ')
```

Output:

```
0 1 1 2 3 5 8 13 21 34
```

6. Write a generator function in Python that yields the powers of 2 up to a given exponent.



```python
for value in powers_of_two(5):
    print(value, end=' ')
```

Output:

```
1 2 4 8 16 32
```

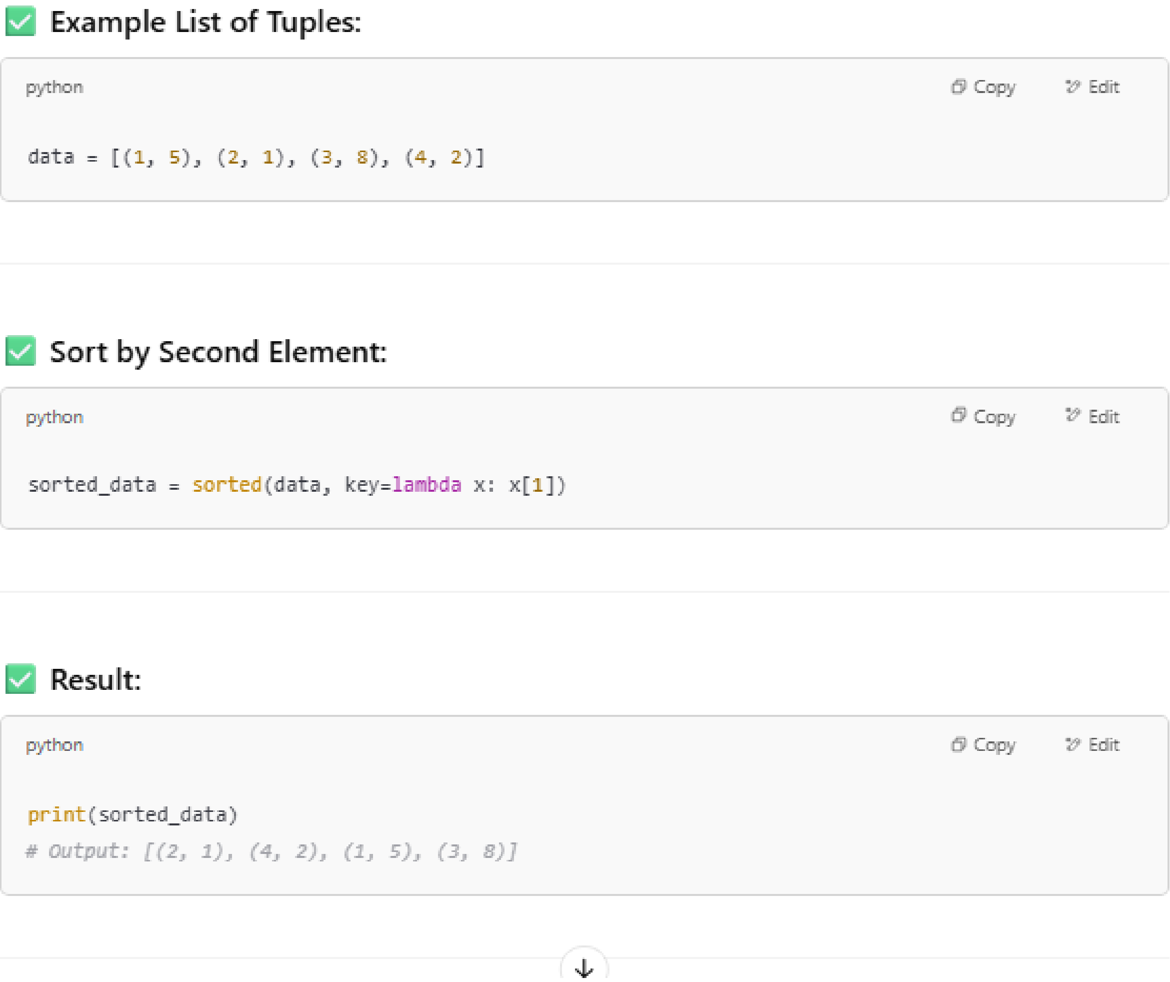
7. Implement a generator function that reads a file line by line and yields each line as a string.

```
def read_file_lines(filename):
    with open(filename, 'r') as file:
        for line in file:
            yield line.rstrip('\n')  # remove trailing newline if needed
```

✅ Example Usage:

```python
for line in read_file_lines('example.txt'):
    print(line)
```

8. Use a lambda function in Python to sort a list of tuples based on the second element of each tuple.



✅ Example List of Tuples:

```python
data = [(1, 5), (2, 1), (3, 8), (4, 2)]
```

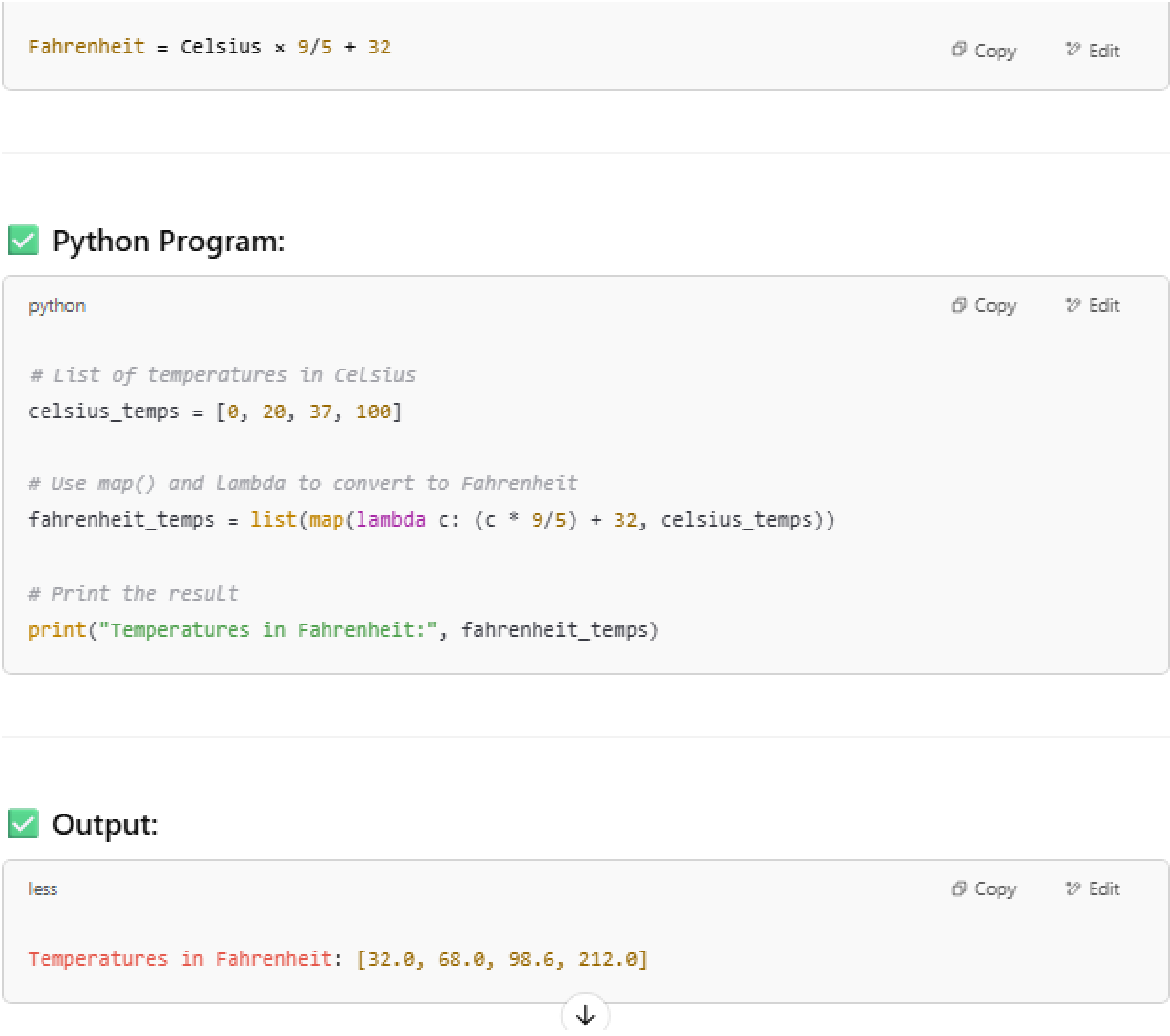
✅ Sort by Second Element:

```python
sorted_data = sorted(data, key=lambda x: x[1])
```

✅ Result:

```python
print(sorted_data)
# Output: [(2, 1), (4, 2), (1, 5), (3, 8)]
```

9. Write a Python program that uses `map()` to convert a list of temperatures from Celsius to Fahrenheit.



```
Fahrenheit = Celsius × 9/5 + 32
```

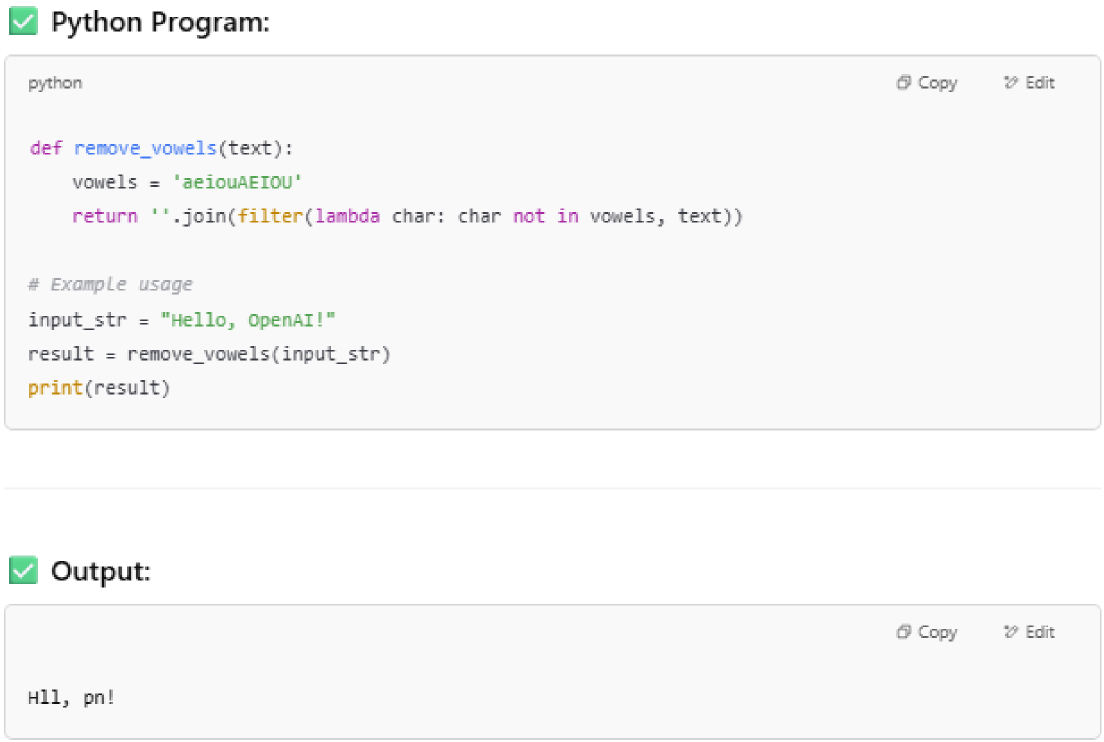
✅ Python Program:

```python
# List of temperatures in Celsius
celsius_temps = [0, 20, 37, 100]

# Use map() and lambda to convert to Fahrenheit
fahrenheit_temps = list(map(lambda c: (c * 9/5) + 32, celsius_temps))

# Print the result
print("Temperatures in Fahrenheit:", fahrenheit_temps)
```

✅ Output:

```
Temperatures in Fahrenheit: [32.0, 68.0, 98.6, 212.0]
```

10. Create a Python program that uses `filter()` to remove all the vowels from a given string.



✅ **Python Program:**

```python
def remove_vowels(text):
    vowels = 'aeiouAEIOU'
    return ''.join(filter(lambda char: char not in vowels, text))

# Example usage
input_str = "Hello, OpenAI!"
result = remove_vowels(input_str)
print(result)
```

✅ **Output:**

```
Hll, pn!
```

11. Imagine an accounting routine used in a book shop. It works on a list with sublists, which look like this: Write a Python program, which returns a list with 2-tuples. Each tuple consists of the order number and the product of the price per item and the quantity. The product should be increased by 10,- € if the value of the order is smaller than 100,00 €. Write a Python program using lambda and map.

```
orders = [
    [1, 20, 3],
    [2, 100, 1],
    [3, 15, 8],
    [4, 5, 10]
]

result = list(map(
    lambda order: (
        order[0],
        order[1] * order[2] + (10 if order[1] * order[2] < 100 else 0)
    ),
    orders
))

print(result)
```

✅ Output:

css Copy Edit

```
[(1, 70), (2, 100), (3, 120), (4, 60)]
```