LAB REPORT # 4&5

NAME : AMISH BABAR

REG NO: B22F1253AI092

SUBJECT: PROGRAMMING OF AI

SECTION: AI BLUE F22

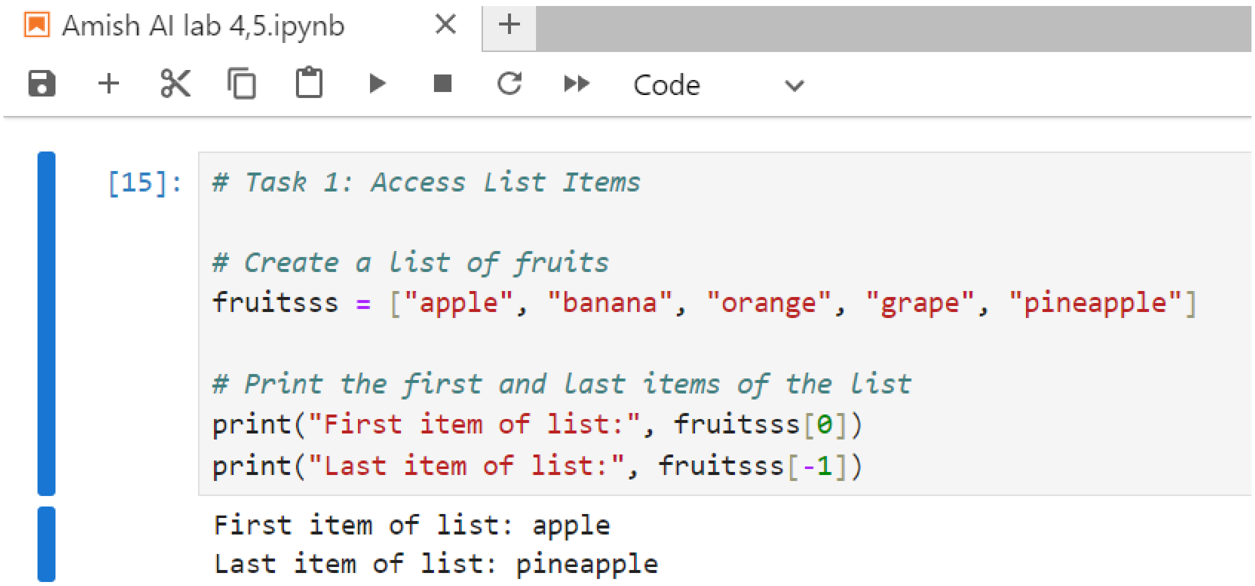
PROGRAM: (AI)

LAB 4 AND 5: PROGRAMMING FOR AI:

Tasks:

Here are some tasks you can design using Python operations and concepts related to lists:

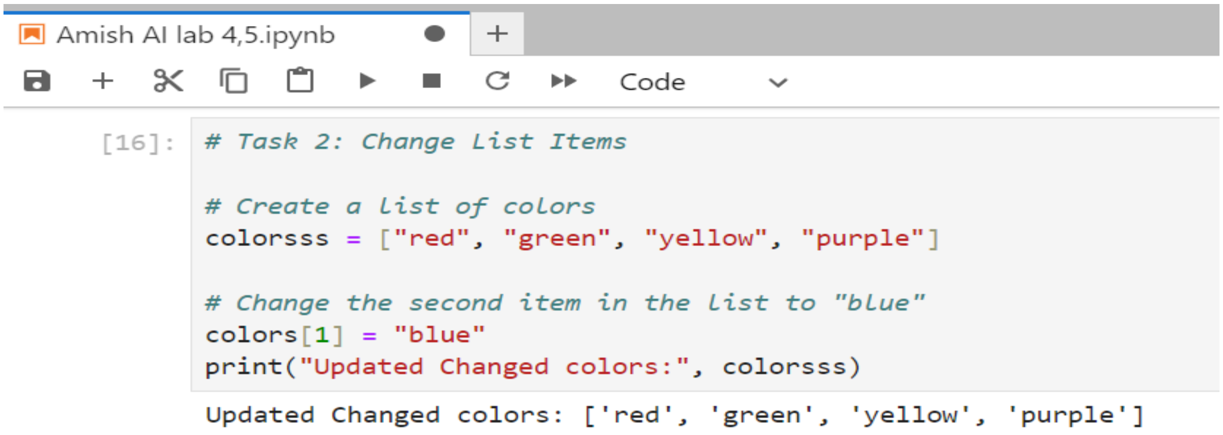
1. **Access List Items:**
   - Create a list of fruits.
   - Print the first and last items of the list.



```
[15]: # Task 1: Access List Items

      # Create a list of fruits
      fruitsss = ["apple", "banana", "orange", "grape", "pineapple"]

      # Print the first and last items of the list
      print("First item of list:", fruitsss[0])
      print("Last item of list:", fruitsss[-1])
```

```
First item of list: apple
Last item of list: pineapple
```

2. **Change List Items:**
   - Create a list of colors.
   - Change the second item in the list to "blue".

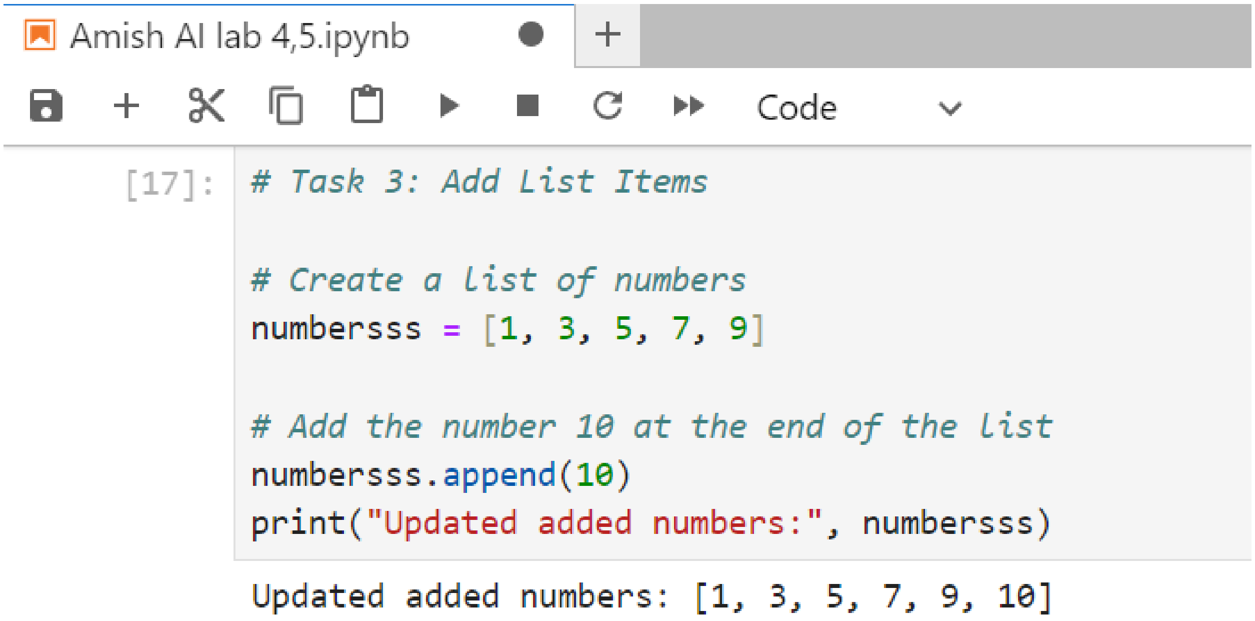


```
[16]: # Task 2: Change List Items

      # Create a list of colors
      colorsss = ["red", "green", "yellow", "purple"]

      # Change the second item in the list to "blue"
      colors[1] = "blue"
      print("Updated Changed colors:", colorsss)
```

```
Updated Changed colors: ['red', 'green', 'yellow', 'purple']
```

3. **Add List Items:**

    - Create a list of numbers.
    - Add the number 10 at the end of the list.

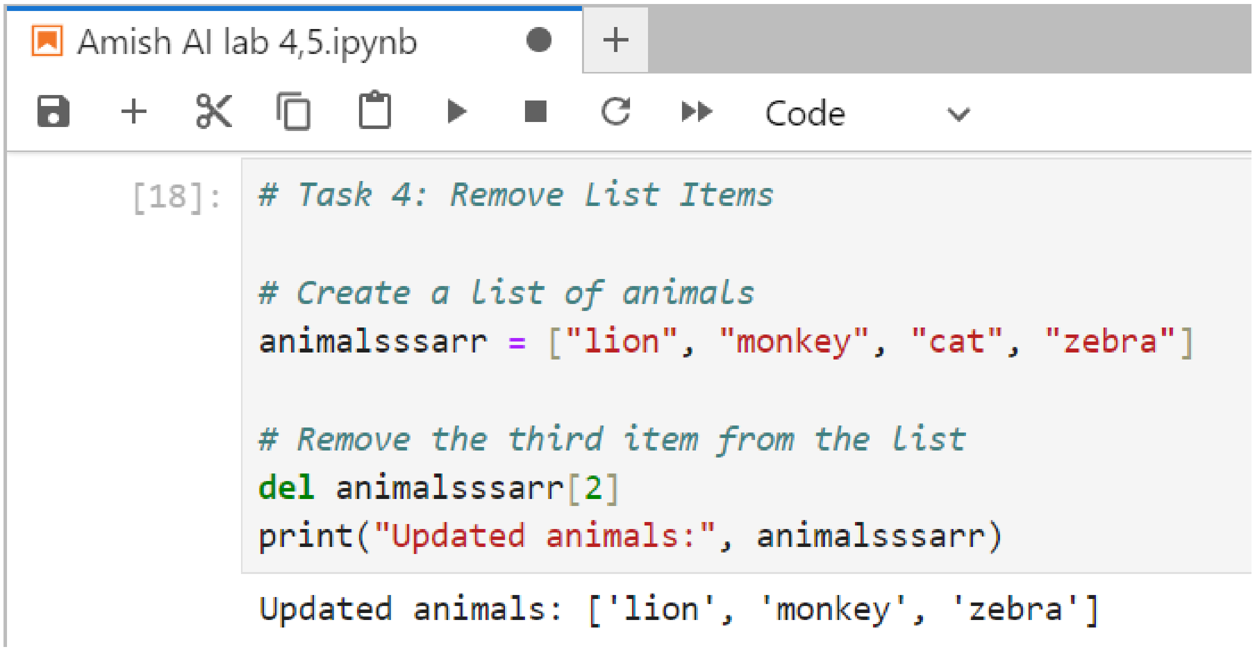


[17]:
```python
# Task 3: Add List Items

# Create a list of numbers
numbersss = [1, 3, 5, 7, 9]

# Add the number 10 at the end of the list
numbersss.append(10)
print("Updated added numbers:", numbersss)
```

```
Updated added numbers: [1, 3, 5, 7, 9, 10]
```

4. **Remove List Items:**

    - Create a list of animals.
    - Remove the third item from the list.

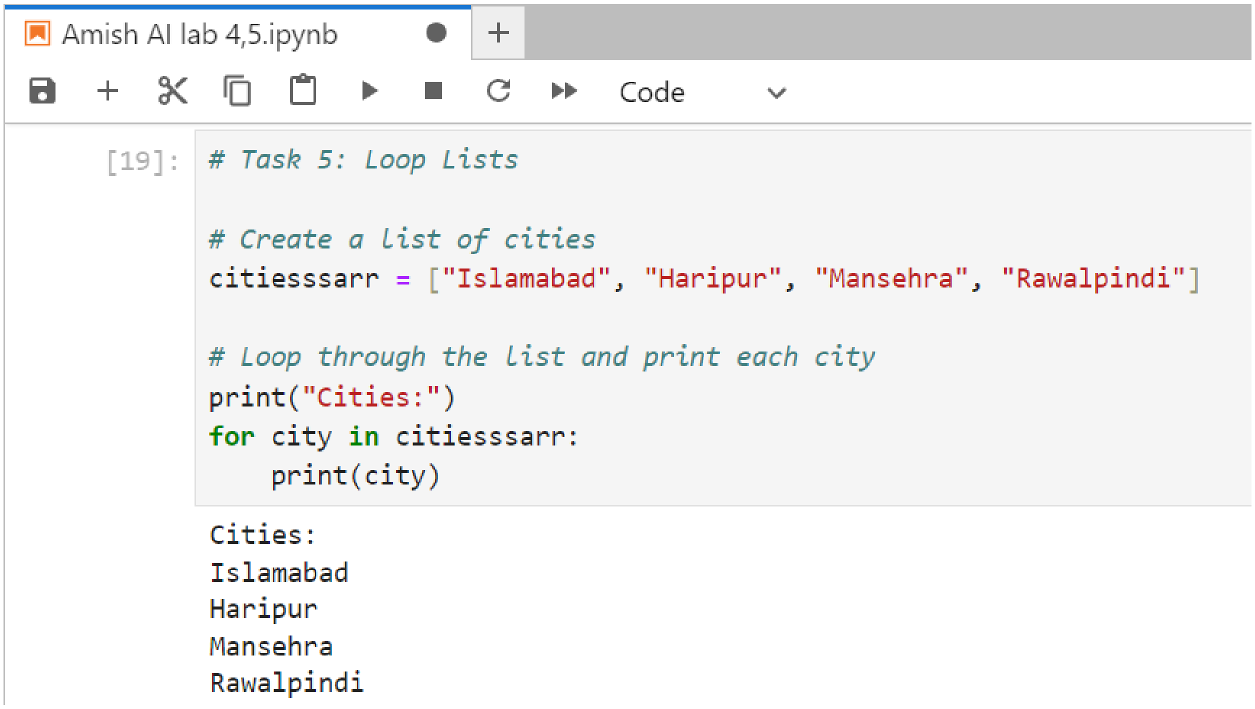


[18]:
```python
# Task 4: Remove List Items

# Create a list of animals
animalsssarr = ["lion", "monkey", "cat", "zebra"]

# Remove the third item from the list
del animalsssarr[2]
print("Updated animals:", animalsssarr)
```

```
Updated animals: ['lion', 'monkey', 'zebra']
```

5. **Loop Lists:**

- Create a list of cities.
- Loop through the list and print each city.

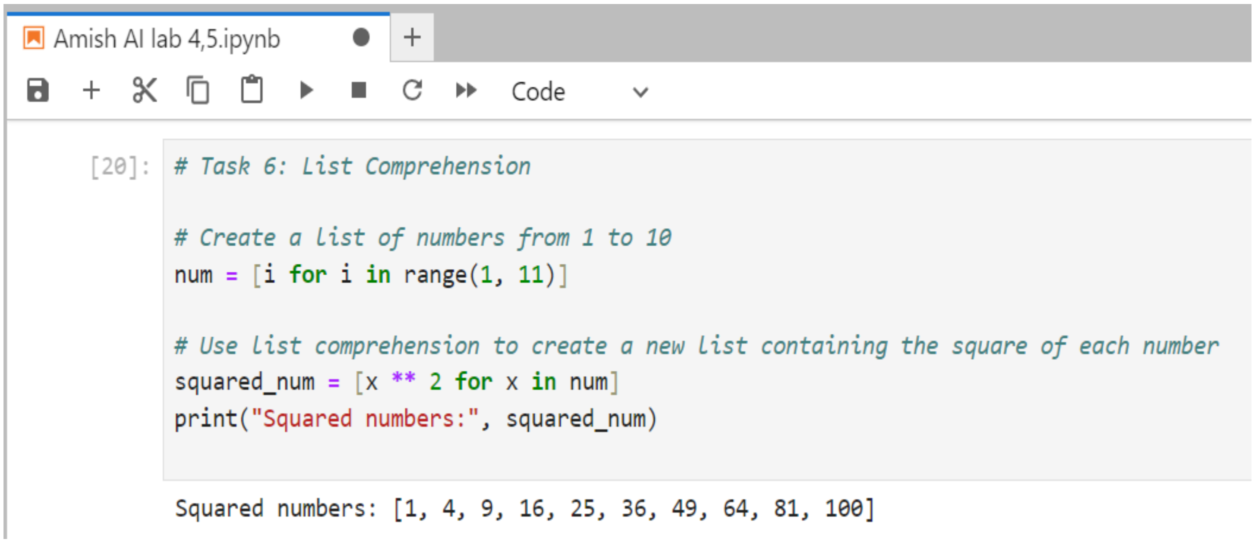


```
# Task 5: Loop Lists

# Create a list of cities
citiesssarr = ["Islamabad", "Haripur", "Mansehra", "Rawalpindi"]

# Loop through the list and print each city
print("Cities:")
for city in citiesssarr:
    print(city)
```

```
Cities:
Islamabad
Haripur
Mansehra
Rawalpindi
```

6. **List Comprehension:**

- Create a list of numbers from 1 to 10.
- Use list comprehension to create a new list containing the square of each number.

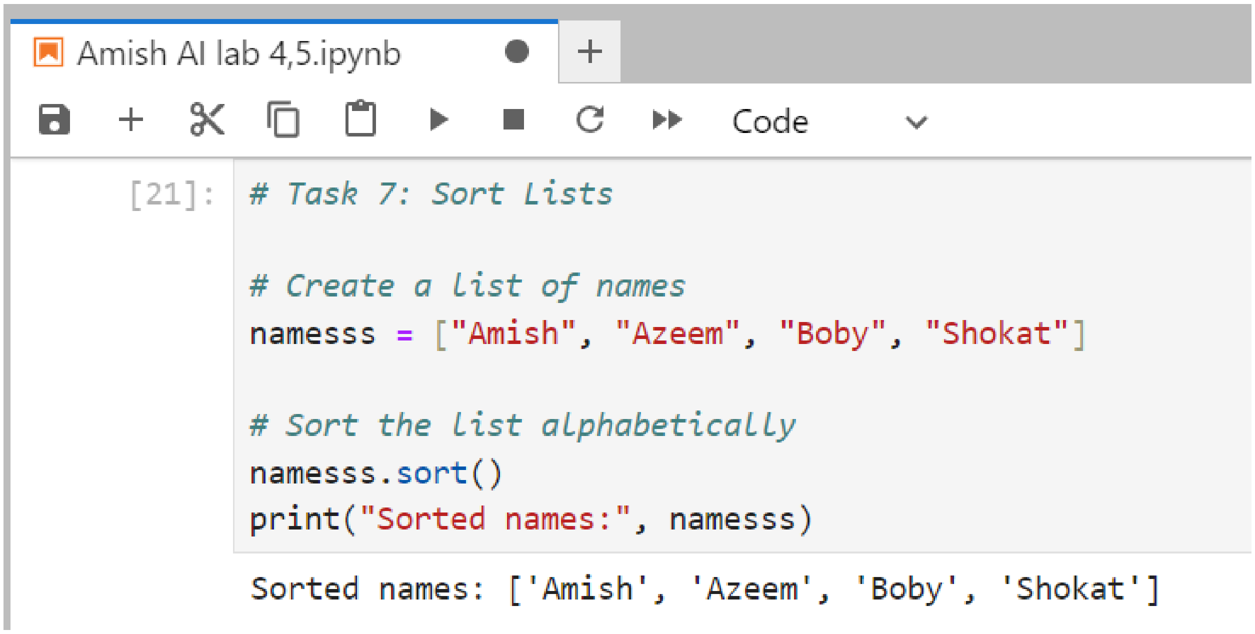


```
# Task 6: List Comprehension

# Create a list of numbers from 1 to 10
num = [i for i in range(1, 11)]

# Use list comprehension to create a new list containing the square of each number
squared_num = [x ** 2 for x in num]
print("Squared numbers:", squared_num)
```

```
Squared numbers: [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

7. **Sort Lists:**

    - Create a list of names.
    - Sort the list alphabetically.

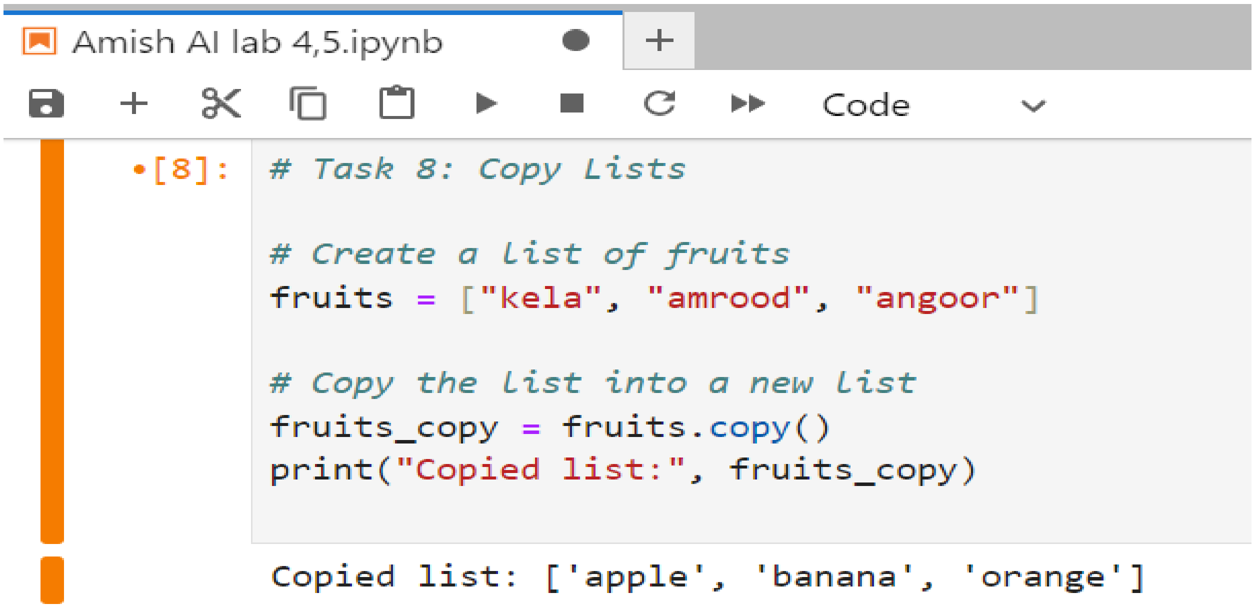


```python
# Task 7: Sort Lists

# Create a list of names
namesss = ["Amish", "Azeem", "Boby", "Shokat"]

# Sort the list alphabetically
namesss.sort()
print("Sorted names:", namesss)
```

```
Sorted names: ['Amish', 'Azeem', 'Boby', 'Shokat']
```

8. **Copy Lists:**

    - Create a list of fruits.
    - Copy the list into a new list.

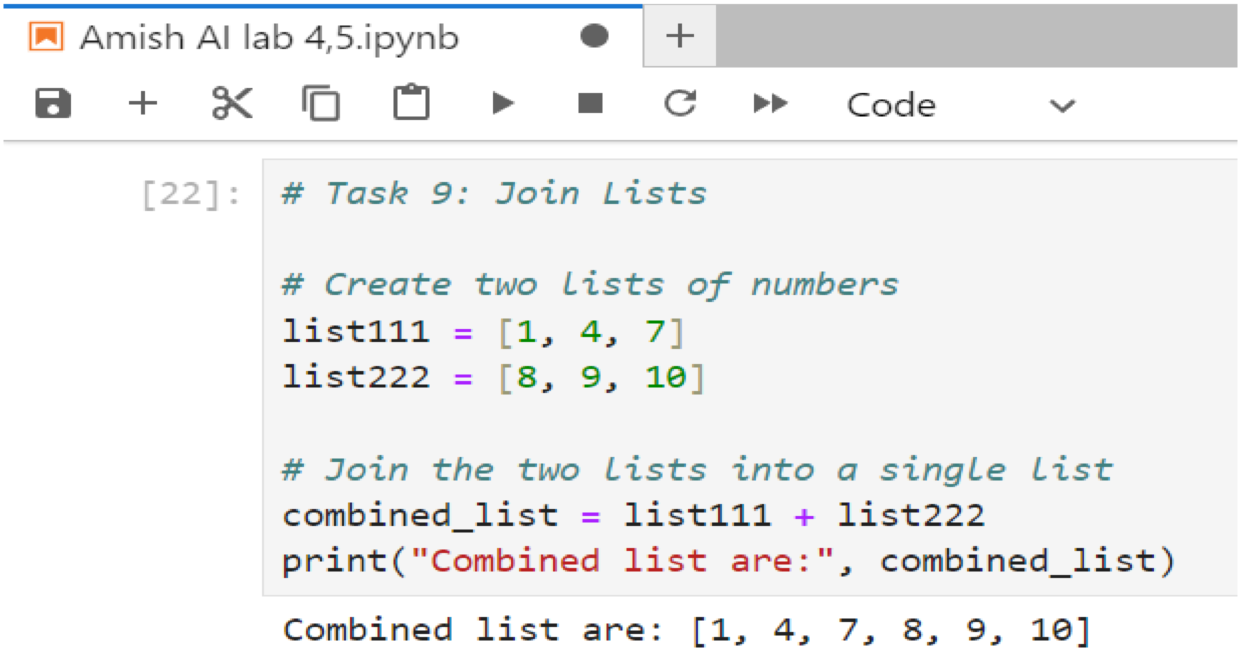


```python
# Task 8: Copy Lists

# Create a list of fruits
fruits = ["kela", "amrood", "angoor"]

# Copy the list into a new list
fruits_copy = fruits.copy()
print("Copied list:", fruits_copy)
```

```
Copied list: ['apple', 'banana', 'orange']
```

9. **Join Lists:**

- Create two lists of numbers.
- Join the two lists into a single list.

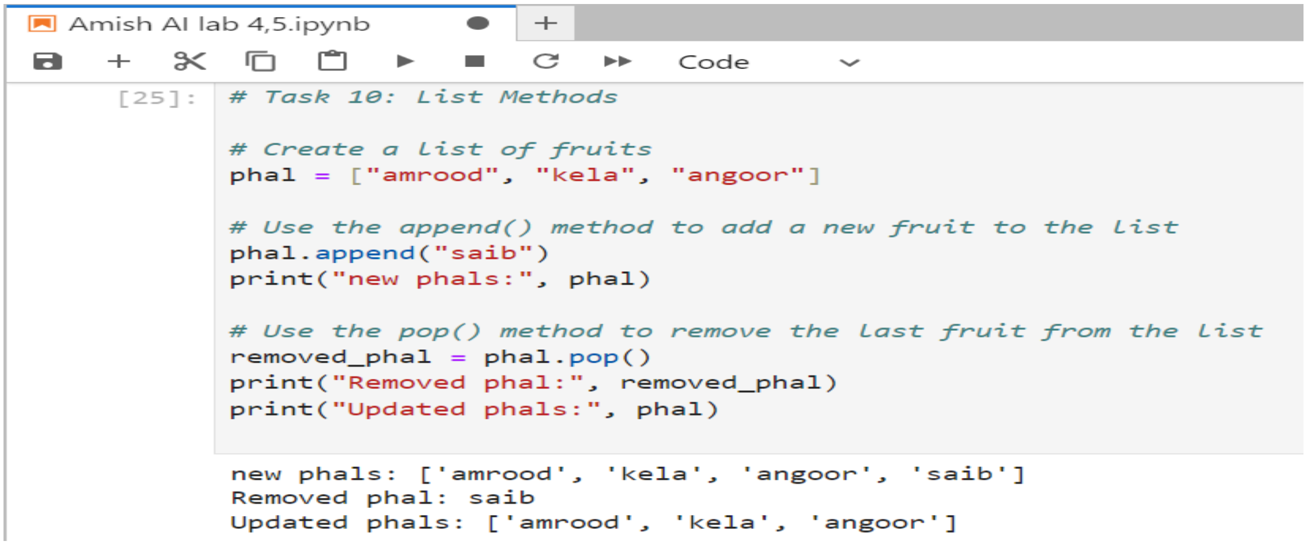


```
[22]: # Task 9: Join Lists

      # Create two lists of numbers
      list111 = [1, 4, 7]
      list222 = [8, 9, 10]

      # Join the two lists into a single list
      combined_list = list111 + list222
      print("Combined list are:", combined_list)
```

```
Combined list are: [1, 4, 7, 8, 9, 10]
```

10. **List Methods:**

- Create a list of fruits.
- Use the **append()** method to add a new fruit to the list.
- Use the **pop()** method to remove the last fruit from the list.

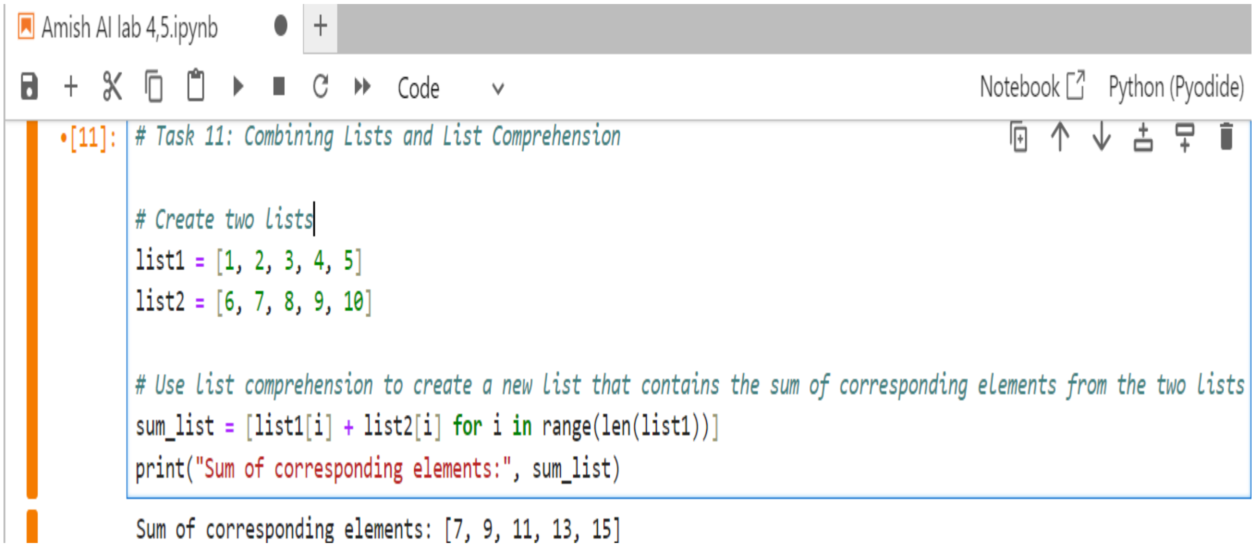


```
[25]: # Task 10: List Methods

      # Create a list of fruits
      phal = ["amrood", "kela", "angoor"]

      # Use the append() method to add a new fruit to the list
      phal.append("saib")
      print("new phals:", phal)

      # Use the pop() method to remove the last fruit from the list
      removed_phal = phal.pop()
      print("Removed phal:", removed_phal)
      print("Updated phals:", phal)
```

```
new phals: ['amrood', 'kela', 'angoor', 'saib']
Removed phal: saib
Updated phals: ['amrood', 'kela', 'angoor']
```

11. Combining Lists and List Comprehension:

- Create two lists, one containing numbers from 1 to 5 and another containing numbers from 6 to 10.
- Use list comprehension to create a new list that contains the sum of corresponding elements from the two lists.



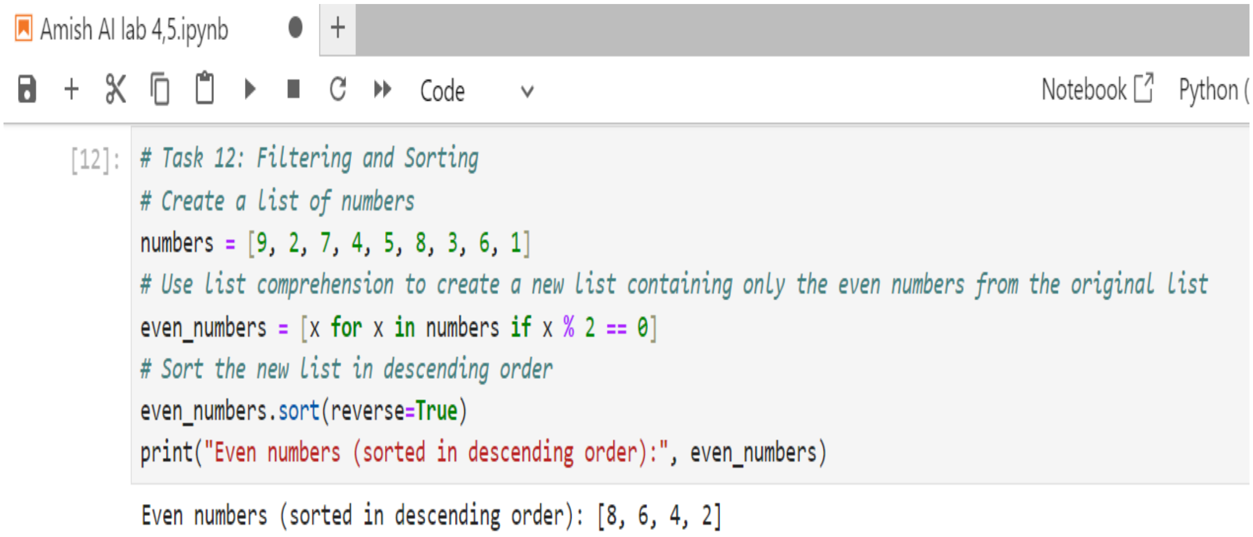
```
# Task 11: Combining Lists and List Comprehension

# Create two lists
list1 = [1, 2, 3, 4, 5]
list2 = [6, 7, 8, 9, 10]

# Use list comprehension to create a new list that contains the sum of corresponding elements from the two lists
sum_list = [list1[i] + list2[i] for i in range(len(list1))]
print("Sum of corresponding elements:", sum_list)
```

```
Sum of corresponding elements: [7, 9, 11, 13, 15]
```

12. Filtering and Sorting:

Create a list of numbers.

- Use list comprehension to create a new list containing only the even numbers from the original list.
- Sort the new list in descending order.

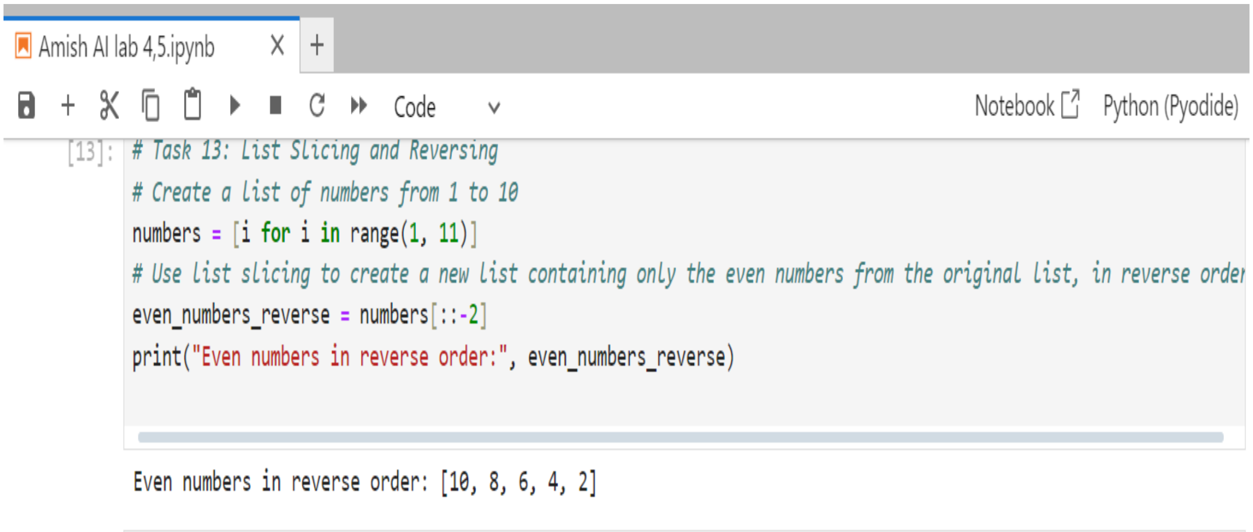


```
# Task 12: Filtering and Sorting
# Create a list of numbers
numbers = [9, 2, 7, 4, 5, 8, 3, 6, 1]
# Use list comprehension to create a new list containing only the even numbers from the original list
even_numbers = [x for x in numbers if x % 2 == 0]
# Sort the new list in descending order
even_numbers.sort(reverse=True)
print("Even numbers (sorted in descending order):", even_numbers)
```

```
Even numbers (sorted in descending order): [8, 6, 4, 2]
```

13. **List Slicing and Reversing:**

- Create a list of numbers from 1 to 10.
- Use list slicing to create a new list containing only the even numbers from the original list, in reverse order.

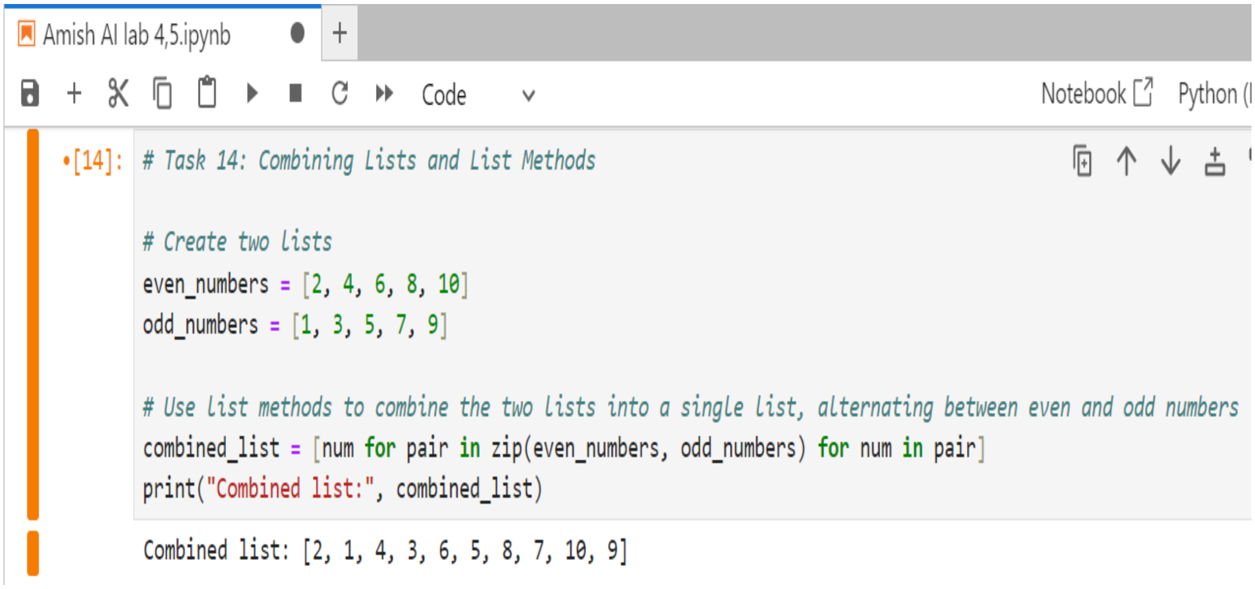


```
# Task 13: List Slicing and Reversing
# Create a list of numbers from 1 to 10
numbers = [i for i in range(1, 11)]
# Use list slicing to create a new list containing only the even numbers from the original list, in reverse orde
even_numbers_reverse = numbers[::-2]
print("Even numbers in reverse order:", even_numbers_reverse)
```

```
Even numbers in reverse order: [10, 8, 6, 4, 2]
```

14. Combining Lists and List Methods:

- Create two lists, one containing even numbers and another containing odd numbers.
- Use list methods to combine the two lists into a single list, alternating between even and odd numbers.



```
# Task 14: Combining Lists and List Methods

# Create two lists
even_numbers = [2, 4, 6, 8, 10]
odd_numbers = [1, 3, 5, 7, 9]

# Use list methods to combine the two lists into a single list, alternating between even and odd numbers
combined_list = [num for pair in zip(even_numbers, odd_numbers) for num in pair]
print("Combined list:", combined_list)
```

```
Combined list: [2, 1, 4, 3, 6, 5, 8, 7, 10, 9]
```