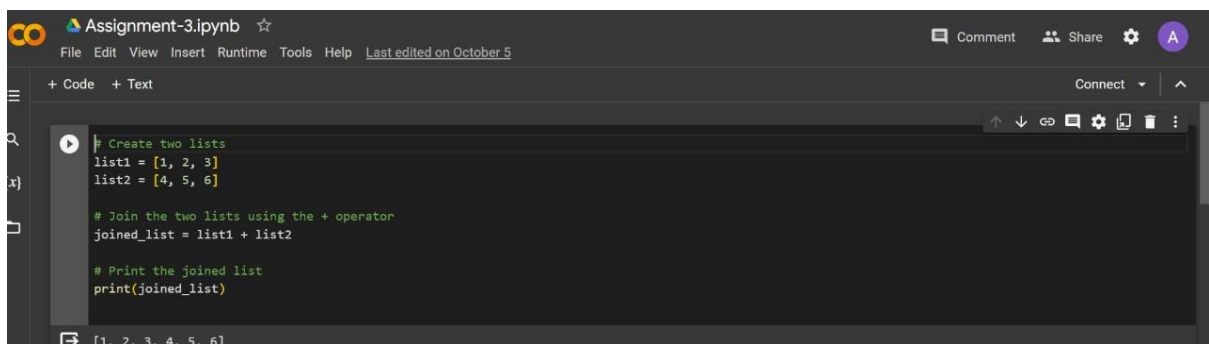


DA ASSIGNMENT – 4

NAME : POLAMREDDY KAVYA

REGISTER NO : 723920104046

Create two list and join those two list :

A screenshot of a Jupyter Notebook titled 'Assignment-3.ipynb'. The code cell contains the following Python code:

```
# Create two lists
list1 = [1, 2, 3]
list2 = [4, 5, 6]

# Join the two lists using the + operator
joined_list = list1 + list2

# Print the joined list
print(joined_list)
```

The output of the code is displayed at the bottom: `[1, 2, 3, 4, 5, 6]`.

```
Assignment-3.ipynb
File Edit View Insert Runtime Tools Help Last edited on October 5
+ Code + Text
# Create two lists
list1 = [1, 2, 3]
list2 = [4, 5, 6]

# Join the two lists using the + operator
joined_list = list1 + list2

# Print the joined list
print(joined_list)
[1, 2, 3, 4, 5, 6]
```

With If statement find the even numbers :

A screenshot of a Jupyter Notebook titled 'Assignment-3.ipynb'. The code cell contains the following Python code:

```
# Create a list of numbers
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

# Create an empty list to store even numbers
even_numbers = []

# Loop through the list and check if each number is even
for num in numbers:
    if num % 2 == 0:
        even_numbers.append(num)

# Print the even numbers
print("Even numbers in the list:", even_numbers)
```

The output of the code is displayed at the bottom: `Even numbers in the list: [2, 4, 6, 8, 10]`.

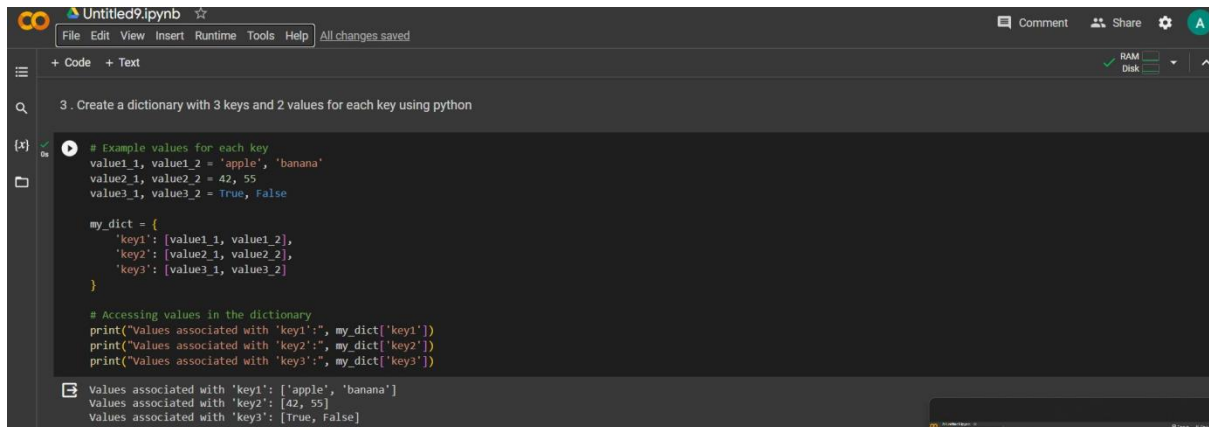
```
Assignment-3.ipynb
File Edit View Insert Runtime Tools Help Last edited on October 5
+ Code + Text
# Create a list of numbers
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

# Create an empty list to store even numbers
even_numbers = []

# Loop through the list and check if each number is even
for num in numbers:
    if num % 2 == 0:
        even_numbers.append(num)

# Print the even numbers
print("Even numbers in the list:", even_numbers)
Even numbers in the list: [2, 4, 6, 8, 10]
```

Create a dictionary with 3 keys and 2 values for each key :



The screenshot shows a Jupyter Notebook interface with a dark theme. The title bar reads 'Untitled9.ipynb'. The menu bar includes 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. Below the menu bar, there are tabs for '+ Code' and '+ Text'. The main code cell contains the following Python code:

```
# Example values for each key
value1_1, value1_2 = 'apple', 'banana'
value2_1, value2_2 = 42, 55
value3_1, value3_2 = True, False

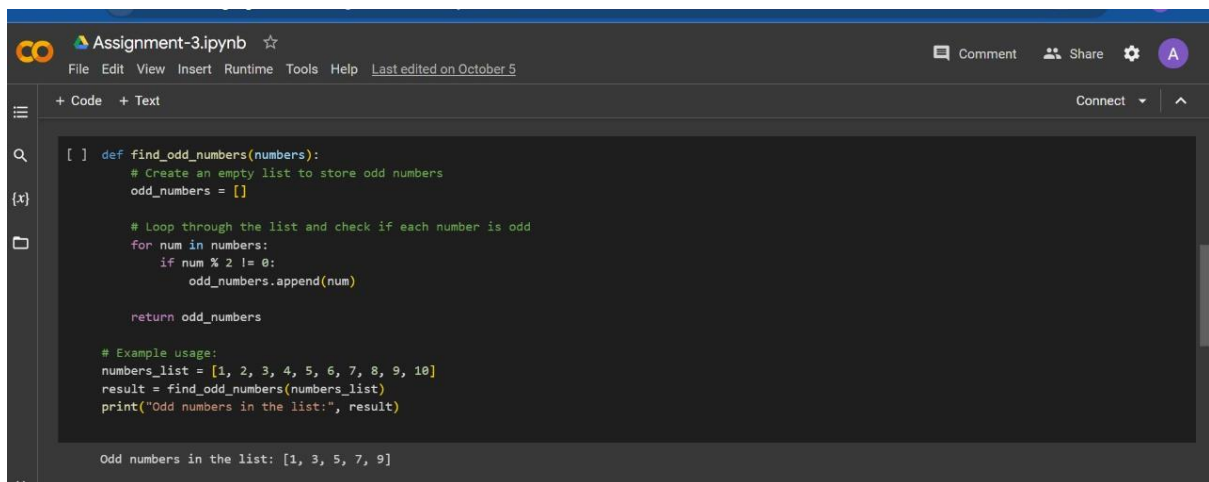
my_dict = {
    'key1': [value1_1, value1_2],
    'key2': [value2_1, value2_2],
    'key3': [value3_1, value3_2]
}

# Accessing values in the dictionary
print("Values associated with 'key1':", my_dict['key1'])
print("Values associated with 'key2':", my_dict['key2'])
print("Values associated with 'key3':", my_dict['key3'])
```

The output of the code is displayed below the code cell:

```
Values associated with 'key1': ['apple', 'banana']
Values associated with 'key2': [42, 55]
Values associated with 'key3': [True, False]
```

Create a function with If statement which is used to find the odd numbers :



The screenshot shows a Jupyter Notebook interface with a dark theme. The title bar reads 'Assignment-3.ipynb'. The menu bar includes 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. Below the menu bar, there are tabs for '+ Code' and '+ Text'. The main code cell contains the following Python code:

```
[ ] def find_odd_numbers(numbers):
    # Create an empty list to store odd numbers
    odd_numbers = []

    # Loop through the list and check if each number is odd
    for num in numbers:
        if num % 2 != 0:
            odd_numbers.append(num)

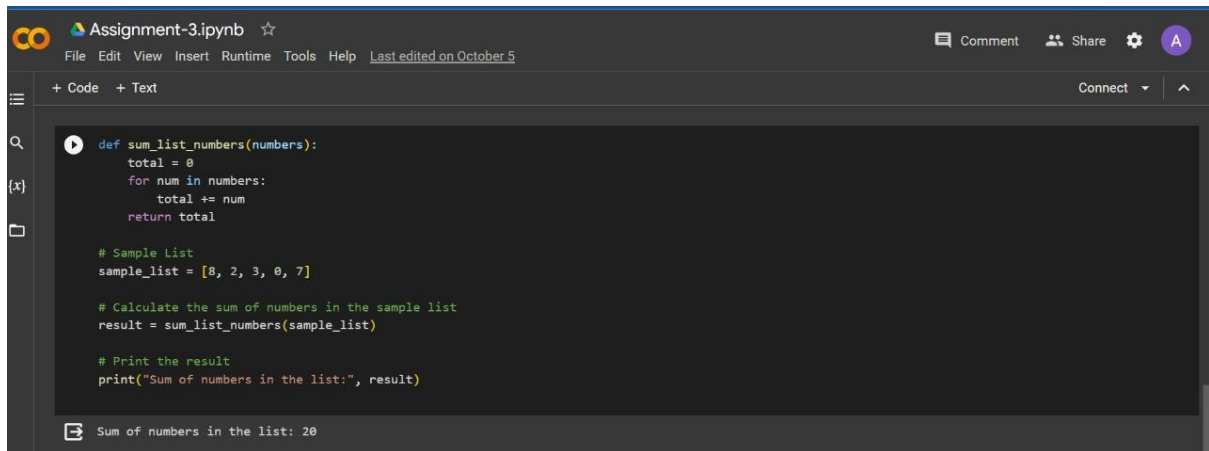
    return odd_numbers

# Example usage:
numbers_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
result = find_odd_numbers(numbers_list)
print("Odd numbers in the list:", result)
```

The output of the code is displayed below the code cell:

```
Odd numbers in the list: [1, 3, 5, 7, 9]
```

Write a Python function to sum all the numbers in a list :



The image shows a Jupyter Notebook interface with a dark theme. The title bar at the top reads "Assignment-3.ipynb" with a star icon on the right. Below the title bar is a menu bar with options: File, Edit, View, Insert, Runtime, Tools, and Help. A status bar on the right of the menu bar says "Last edited on October 5". Below the menu bar is a toolbar with "+ Code" and "+ Text" buttons. On the far right of the toolbar are "Comment", "Share", a settings gear icon, and a user profile icon labeled "A". Below the toolbar is a sidebar with icons for a list, search, a variable {x}, and a file. The main area contains a code cell with a play button icon. The code defines a function `sum_list_numbers(numbers)` that initializes `total = 0`, loops through `numbers` to calculate the sum, and returns `total`. It then creates a `sample_list = [8, 2, 3, 0, 7]`, calls the function, and prints the result. The output at the bottom of the cell is "Sum of numbers in the list: 20".

```
def sum_list_numbers(numbers):
    total = 0
    for num in numbers:
        total += num
    return total

# Sample List
sample_list = [8, 2, 3, 0, 7]

# Calculate the sum of numbers in the sample list
result = sum_list_numbers(sample_list)

# Print the result
print("Sum of numbers in the list:", result)
```

Sum of numbers in the list: 20