Practical-9

March 19, 2025

0.1 Practical 9 :-

Name: Sarthak Sanay
Enrollment No: 230031101611051

0.1.1 Problem Statement 1:-

You are developing a data processing tool where you receive a list of mixed data types. You must filter out non-numeric values and convert the remaining values to integers. Write a Python function `filter_and_convert_to_integers(data)` that takes a list of mixed data types as input, filters out non-numeric values, and returns a new list with the remaining values converted to integers.

[8]:
```
def filter_and_convert_to_integers(data):
    filtered_list = []
    for ele in data:
        if type(ele) == int:
            filtered_list.append(ele)
        if type(ele) == float:
            x = int(ele)
            filtered_list.append(x)
    return filtered_list

data = ["hi", 10, 7.2, 100.55, 51, True, False, 'a', "Sarthak Sanay", 500.
 ↪446789]
print(filter_and_convert_to_integers(data))
```

```
[10, 7, 100, 51, 500]
```

0.1.2 Problem Statement 2:-

You're building a music playlist manager and need to merge two playlists into one sorted playlist. Write a Python function `merge_and_sort_playlists(playlist1, playlist2)` that takes two lists of songs as input, merges them into one list, and sorts the merged list alphabetically.

[9]:
```
def merge_and_sort_playlists(playlist1, playlist2):
    playlist1.extend(playlist2)
    playlist1.sort()
    return playlist1
```

```
playlist1 = ['Song 1', 'Song 2', 'Song 3']
playlist2 = ['Song 4', 'Song 5', 'Song 6']
merged_playlist = merge_and_sort_playlists(playlist1, playlist2)
print("Merged Playlist:", merged_playlist)
```

```
Merged Playlist: ['Song 1', 'Song 2', 'Song 3', 'Song 4', 'Song 5', 'Song 6']
```

0.1.3 Problem Statement 3:-

You're analyzing sales data and need to group sales amounts by product category and calculate the total sales for each category. Write a Python function `calculate_total_sales(sales_data)` that takes a list of tuples (product, amount) as input, where the product is the product category and the amount is the sales amount. It should return a dictionary where keys are product categories and values are the total sales amounts for each category.

[4]:
```
def calculate_total_sales(sales_data):
    dictionary = {}
    for data in sales_data:
        if data[0] not in dictionary:
            dictionary[data[0]] = data[1]
        else:
            dictionary[data[0]] += data[1]
    return dictionary

sales_data = [("electronics", 8000), ("toys", 450), ("groceries", 1250),␣
 ↪("clothing", 2400), ("furniture", 24500), ("toys", 1650), ("groceries",␣
 ↪750), ("electronics", 4500), ("clothing", 3450)]
print( calculate_total_sales(sales_data))
```

```
{'electronics': 12500, 'toys': 2100, 'groceries': 2000, 'clothing': 5850,
'furniture': 24500}
```

0.1.4 Problem Statement 4:-

You're implementing a game where players can rotate a list of numbers to the left by a given number of positions. Write a Python function `rotate_list_left(numbers, positions)` that takes a list of numbers and the number of positions to rotate as input, and returns a new list with the elements rotated to the left by the specified number of positions.Z

[8]:
```
def rotate_list_left(numbers, positions):
    positions = positions % len(numbers)  # To handle cases where positions >␣
 ↪len(numbers)

    for i in range(positions):
        temp0 = numbers[0]
        for j in range(len(numbers) - 1):
            numbers[j] = numbers[j + 1]
        numbers[-1] = temp0  # To move the first element to the end
```

```
    return numbers

numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9]
print(rotate_list_left(numbers, 3))
```

```
[4, 5, 6, 7, 8, 9, 1, 2, 3]
```