Python Programming
Dict Homework 1

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

Problem #1: Special String Mapping

- Read a string and do the following conversions for its letters
  - If it is lower letter, use this map of 26 letters:
    - **a**bcdefghijklmnopqrstuvwxy**z**
    - YZIMNESTODUAPWXHQFBRJKCGVL
    - E.g. a ⇒ Y   and z ⇒ L
  - If it is digit, use this map of 10 letters:
    - 0123456789
    - !@#$%^&*()
  - Otherwise, don't change it!
- Input ⇒ Output
  - acMNmn39 ⇒ YIMNPW$)
  - vwXYZ0123 ⇒ KCXYZ!@#$

Problem#2: Sort by type

- Implement function: `def sort_different_types(lst):`
- It takes a list of different data types (int, float, string, list, tuple)
  - Return a list:
  - For every data type, group and sort from small to large
  - Within data types: Order them based on the order of the list (int comes first? Be first)
  - Assumption: provided lists or tuples will be comparable.

```
lst = [10, 'most', 2.5, 7, 'aly', 9, 4.5, 2, 'ziad', -4, 1.1, [1, 5], 5, [0, 7, 8]]
print(sort_different_types(lst))
```

```
[-4, 2, 5, 7, 9, 10, 'aly', 'most', 'ziad', 1.1, 2.5, 4.5, [0, 7, 8], [1, 5]]
```

Problem #3: Fast Prefix Finder

- Read an Integer N on a line, then read N lines of strings (database)
- Then read an Integer Q on a line, then read Q lines of strings (queries)
- For every query, list all database strings that starts with this query
  - However the database is huge: using query in lst is very slow
- Print entries according to reading order
- See screenshoot

```
5
Hey
Hello
Hi
Mostafa Saad
Mostafa Ziad
4
He
He matches ['Hey', 'Hello']
H
H matches ['Hey', 'Hello', 'Hi']
Most
Most matches ['Mostafa Saad', 'Mostafa Ziad']
MOST
Not found
```

Problem #4: Filter Duplicates!

- Write function: `def filter_duplicates_preserve_order(lst):`
  - Input is list of list of integers
  - Output: A new list after removing all duplicate lists
  - You must **preserve the input order**
  - Use a Dict

```
def filter_duplicates_preserve_order(lst):...

if __name__ == '__main__':
    print(filter_duplicates_preserve_order([[7, 1], [2, 4],
                                            [7, 1], [5, 2], [2, 4]]))
```

```
[[7, 1], [2, 4], [5, 2]]
```

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."