
LIST – Exercises (5)

Exercise 1

Given a list of integers, rearrange the list so that all negative numbers appear first and all positive numbers appear later, without using additional predefined functions like `sort()` .

Exercise 2

Given a list, create a new list that contains only those elements which appear more than once. Do not use `set()` .

Example:

Input: [1, 2, 3, 2, 4, 1, 5, 1]

Output: [1, 2]

Exercise 3

Rotate a list to the left by N positions using only loops.

Exercise 4

Given a list of strings, remove all strings whose length is less than 3.

Exercise 5

Flatten a nested list using loops only.

Example:

Input: [[1, 2], [3, 4], [5]]

Output: [1, 2, 3, 4, 5]

TUPLE – Exercises (5)

Exercise 1

Given a tuple containing both numbers and strings, separate them into two tuples: one containing only numbers and one containing only strings.

Exercise 2

Given a tuple of numbers, determine whether it is strictly increasing.

Exercise 3

Create a tuple of words and return a new tuple where each word is reversed.

Input: ("python", "cloud", "data")

Output: ("nohtyp", "duolc", "atad")

Exercise 4

Write a program to find the index positions of a given value inside a tuple, without using `index()`.

Exercise 5

Given a tuple `(10, (20, 30), (40, (50, 60)))`, print all integers using recursion.

SET – Exercises (5)

Exercise 1

Given two sets, print the elements that are present in the union but not in the intersection.
(Equivalent to symmetric difference but do not use `^`.)

Exercise 2

Given a list with duplicates, convert it to a set and then back to a list, preserving the original order of first occurrences.

Exercise 3

Given a set of numbers, find all pairs that sum up to a target number.

Example:

Set = {2, 4, 6, 8, 10}

Target = 12

Output: (2, 10), (4, 8), (6, 6)

Without reusing same pairs twice.

Exercise 4

Check whether two sets are disjoint without using the built-in method.

Exercise 5

Given a list of words, find all unique characters across all words combined, using sets.

DICTIONARY – Exercises (5)

Exercise 1

Given a dictionary of student marks:

{ "A": 85, "B": 92, "C": 78, "D": 92 }

Print all keys that have the maximum value without using `max()`.

Exercise 2

Invert a dictionary (swap key and value).

If duplicate values exist, group keys into a list.

Input: {"a": 1, "b": 2, "c": 1}

Output: {1: ['a', 'c'], 2: ['b']}

Exercise 3

Merge two dictionaries, but if a key appears in both, store the values in a list.

Input: {a: 1, b: 2} and {b: 3, c: 4}

Output: {a: 1, b: [2, 3], c: 4}

Exercise 4

Given a dictionary representing items and prices, remove all items whose price is below 100.

Exercise 5

Given a nested dictionary of employees, count how many employees belong to each department.

Example input:

```
{  
    "e1": {"dept": "IT"},  
    "e2": {"dept": "HR"},  
    "e3": {"dept": "IT"},  
}
```

Output:

```
{"IT": 2, "HR": 1}
```
