

# INTERNSHIP TASKS

## Day 13: EXERCISE – 13

Name	: S. Deva Manikanta
Clg Id	: 12119003
Course	: Python
Org	: IGIAT – VSKP
Date	: 12-04-2024

## Exercise Level 1

```
#Task 1:
#Filter only negative and zero in the list using list comprehension
def only_postives(list_of_values):
    updated_list = [value for value in list_of_values if value != 0 and value > 0]
    return updated_list;
print("Task 1: List with out negative and zero values --> RESULT AS FOLLOWS");
print("List 1 -- [-4, -3, -2, -1, 0, 2, 4, 6]: ", only_postives([-4, -3, -2, -1, 0, 2, 4, 6]));
print("List 2 -- [0, -1, 2, 0, 4, 5, -2]: ", only_postives([0, -1, 2, 0, 4, 5, -2]));

#Task 2:
#Flatten the following list of lists of lists to one dimensional list;
list_of_lists = [[[1, 2, 3]], [[4, 5, 6]], [[7, 8, 9]]]
list_of_values = [list_item_of_list for list in list_of_lists for list_item in list for list_item_of_list in list_item];
print("Task 2: [[[1, 2, 3]], [[4, 5, 6]], [[7, 8, 9]]] --(flatten)-->",list_of_values);

#Task 3:
#Using list comprehension create the following list of tuples in 13.List Comprehension.pptx
output_task_3 = [(i, i**0, i**1, i**2, i**3, i**4, i**5) for i in range(0, 11)];
print("Task 3: Output As follows\n");
print("")
for i in output_task_3:
    print(" ",i);
print("");

#Task 4:
#Flatten the following list to a new list
# [(['Finland', 'Helsinki']), (['Sweden', 'Stockholm']), (['Norway', 'Oslo'])]
list_of_countries = [(['Finland', 'Helsinki']), (['Sweden', 'Stockholm']), (['Norway', 'Oslo'])];
country_codes = {
    'Finland' : 'FIN', 'Sweden' : 'SWE', 'Norway' : 'NOR'
}
output_task_4 = [[country.upper(), country_codes[country], location.upper()] for (country, location) in list_of_countries];
print("\nTask 4: Result as follows:\n", output_task_4);

#Task 5:
# Change the list of tuples to a dictionary
list_of_countries = [(['Finland', 'Helsinki']), (['Sweden', 'Stockholm']), (['Norway', 'Oslo'])];
output_task_5 = [{'country': country_name, 'city' : city_name} for (country_name, city_name) in list_of_countries]
print("\nTask 5: Result as follows:");
print(output_task_5);

#Task 6:
# Change the list of lists to a list of concatenated strings
```

```

names = [[('Asabeneh', 'Yetayeh')], [('David', 'Smith')], [('Donald', 'Trump')], [('Bill', 'Gates')]]
output_task_6 = [first_name+" "+last_name for [(first_name, last_name)] in names];
print("\nTask 6: Output as follows");
print(output_task_6);

#Task 7:
#Write a lambda function which can solve a slope or y-intercept of linear functions
equation = "y = 2x + 3"
def slope_intercept_finder(equation):
    equation = equation.replace(" ", '');
    equation_parts = equation.split("=")
    slope_y_intercept_finder = lambda equation : {'slope(m)' :
int(equation[1][0:equation[1].index('x')].strip()), 'y-intercept(b)' :
int(equation[1][equation[1].index('x')+1:].strip())}
    return slope_y_intercept_finder(equation_parts);

output_task_7 = slope_intercept_finder("y = 2x + 3");
print("\nTask 7 --> Linear Function => y = mx+b, HERE: 2x+3")
print("Slope [m]: ", output_task_7['slope(m)'], "\nY-intercept [b]:", output_task_7['y-intercept(b)'])

```

## Outputs:

```

@DevaManikantaSala →/workspaces/codespaces-blank $ /home/codespace/.python/current/bin/python3 "/workspaces/codespaces-blank/IGIAT Internship Python Tasks/30DaysOfPython/day_13/exercise.py"
Task 1: List with out negative and zero values --> RESULT AS FOLLOWS
List 1 -- [-4, -3, -2, -1, 0, 2, 4, 6]: [2, 4, 6]
List 2 -- [0, -1, 2, 0, 4, 5, -2]: [2, 4, 5]
Task 2: [[1, 2, 3]], [[4, 5, 6]], [[7, 8, 9]] --(flatten)--> [1, 2, 3, 4, 5, 6, 7, 8, 9]
Task 3: Output As follows

[
  (0, 1, 0, 0, 0, 0, 0)
  (1, 1, 1, 1, 1, 1, 1)
  (2, 1, 2, 4, 8, 16, 32)
  (3, 1, 3, 9, 27, 81, 243)
  (4, 1, 4, 16, 64, 256, 1024)
  (5, 1, 5, 25, 125, 625, 3125)
  (6, 1, 6, 36, 216, 1296, 7776)
  (7, 1, 7, 49, 343, 2401, 16807)
  (8, 1, 8, 64, 512, 4096, 32768)
  (9, 1, 9, 81, 729, 6561, 59049)
  (10, 1, 10, 100, 1000, 10000, 100000)
]

Task 4: Result as follows:
[['FINLAND', 'FIN', 'HELSINKI'], ['SWEDEN', 'SWE', 'STOCKHOLM'], ['NORWAY', 'NOR', 'OSLO']]

Task 5: Result as follows:
[{'country': 'Finland', 'city': 'Helsinki'}, {'country': 'Sweden', 'city': 'Stockholm'}, {'country': 'Norway', 'city': 'Oslo'}]

Task 6: Output as follows
['Asabeneh Yetayeh', 'David Smith', 'Donald Trump', 'Bill Gates']

Task 7 --> Linear Function => y = mx+b, HERE: 2x+3
Slope [m]: 2
Y-intercept [b]: 3
@DevaManikantaSala →/workspaces/codespaces-blank $

```