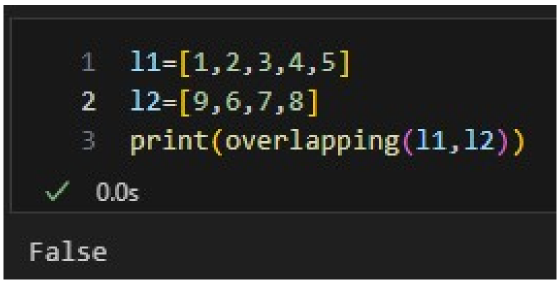
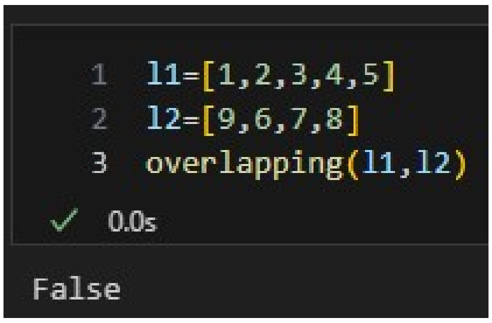
Assignment 3

Q1. Define a function overlapping () that takes two lists and returns True if they have at least one member in common, False otherwise.



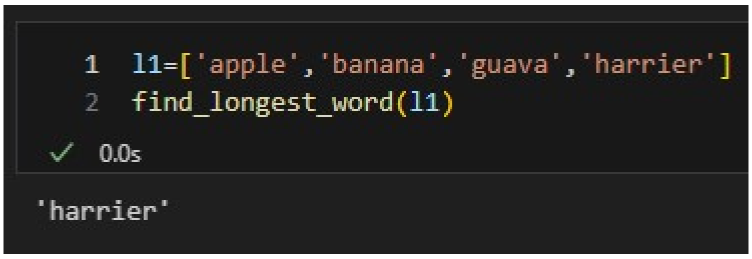
```
l1=[1,2,3,4,5]
l2=[9,6,7,8]
print(overlapping(l1,l2))
✓ 0.0s

False
```



```
l1=[1,2,3,4,5]
l2=[9,6,7,8]
overlapping(l1,l2)
✓ 0.0s

False
```

```
def overlapping(l1,l2):
    len1=len(l1)
    len2=len(l2)
    s=set(l1+l2)
    len3=len(s)

    if len3!=len1+len2:
        return True
    elif len3==len1+len2:
        return False
```

Q.2. Write a function find_longest_word() to find the longest word from the list of words



```
l1=['apple','banana','guava','harrier']
find_longest_word(l1)
✓ 0.0s

'harrier'
```

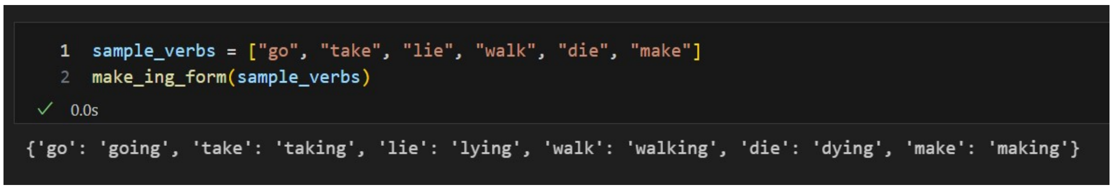
```
def find_longest_word(l1):
    longest_word=''
    for i in l1:
        if len(i)>len(longest_word):
            longest_word=i
    return longest_word
```

Q.3.In English, present participle is formed by adding suffix -ing to infinite form: go -> going. A simple set of rules can be given as follows:

a. If the verb ends in e, drop the e and add ing

b. If the verb ends in ie, change ie to y and add ing

Write a function make_ing_form() which accepts a list of verbs and returns a dictionary with verb : present participle



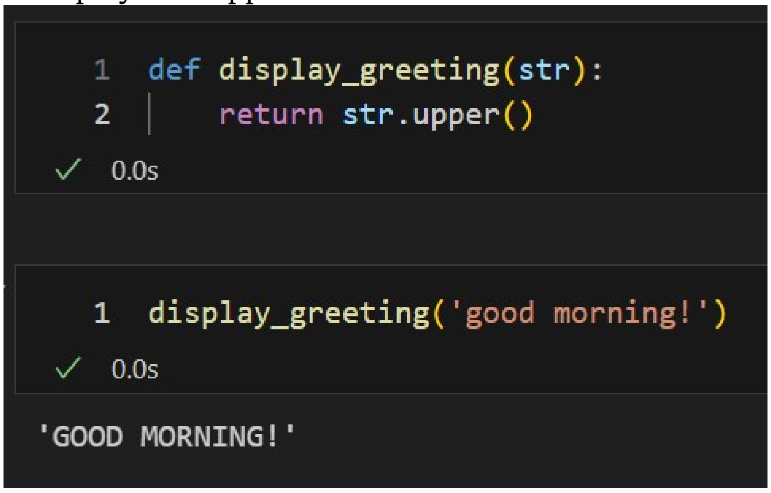
```
sample_verbs = ["go", "take", "lie", "walk", "die", "make"]
make_ing_form(sample_verbs)
```

✓ 0.0s

```
{'go': 'going', 'take': 'taking', 'lie': 'lying', 'walk': 'walking', 'die': 'dying', 'make': 'making'}
```

```
def make_ing_form(l1):
    d1={}
    for i in l1:
        word=i
        if word[-1]=='e':
            word=word[:-1]
        if word[-1]=='i':
            word=word[:-1]+'y'
        word=word+'ing'
        d1[i]=f'{word}'
    print(d1)
```

Q.4. Decorate the display_greeting() function using a decorator so that greeting is displayed in uppercase



```
def display_greeting(str):
    return str.upper()
```

✓ 0.0s

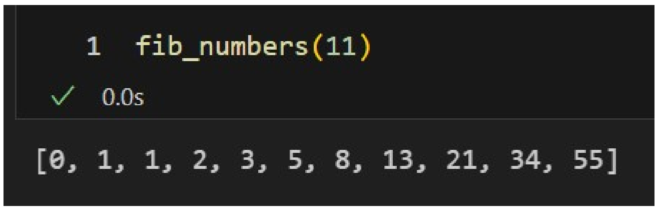
```
display_greeting('good morning!')
```

✓ 0.0s

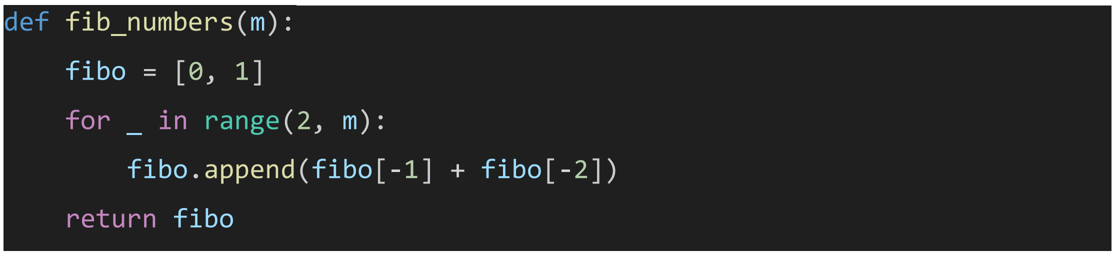
```
'GOOD MORNING!'
```

```
def display_greeting(str):
    return str.upper()
```

Q. 5. Create a infinite series on fib numbers and print first few



```
fib_numbers(11)
✓ 0.0s
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55]
```



```
def fib_numbers(m):
    fibo = [0, 1]
    for _ in range(2, m):
        fibo.append(fibo[-1] + fibo[-2])
    return fibo
```

Q.6

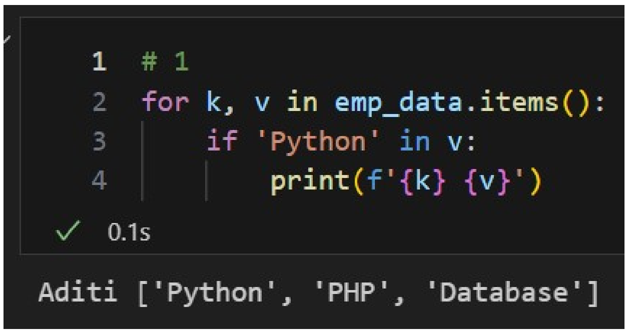
1.Find employees that know 'python'

2. Add a new skill - 'test' in skillset of all employees

3. Sort employees by skills

for the given dictionary of employees

emp_data = {'Amol': ['C', 'C++', 'Java'], 'Aditya': ['Angular', 'Java'],
 'Aditi': ['Python', 'PHP', 'Database']}

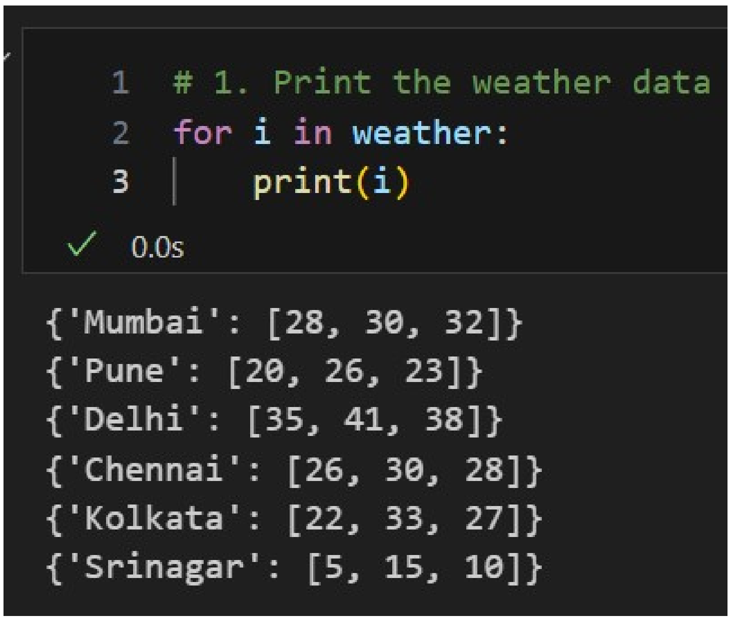


```
# 1
for k, v in emp_data.items():
    if 'Python' in v:
        print(f'{k} {v}')
✓ 0.1s
Aditi ['Python', 'PHP', 'Database']
```

Q.7 Following data displays min/max/average temp for cities

weather= [{'Mumbai' : [28, 30, 32]},.....]

1. Print the weather data

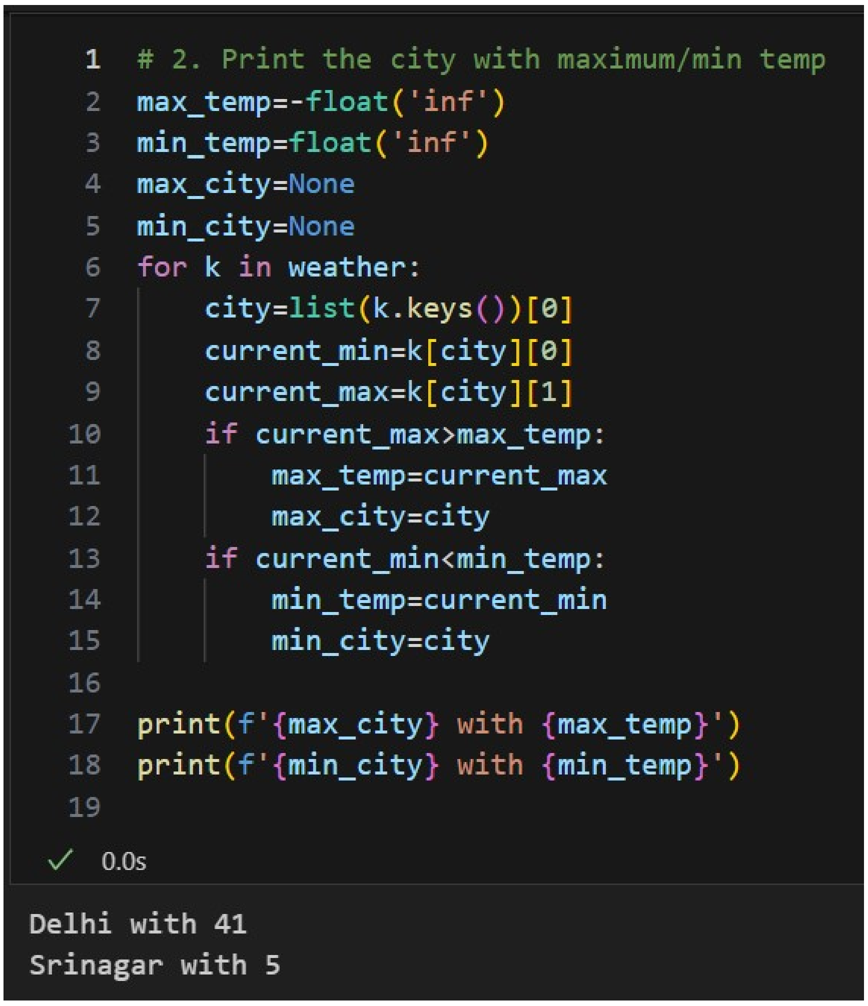


```python
# 1. Print the weather data
for i in weather:
    print(i)
```

✓ 0.0s

```
{'Mumbai': [28, 30, 32]}
{'Pune': [20, 26, 23]}
{'Delhi': [35, 41, 38]}
{'Chennai': [26, 30, 28]}
{'Kolkata': [22, 33, 27]}
{'Srinagar': [5, 15, 10]}
```

2. Print the city with maximum/min temp

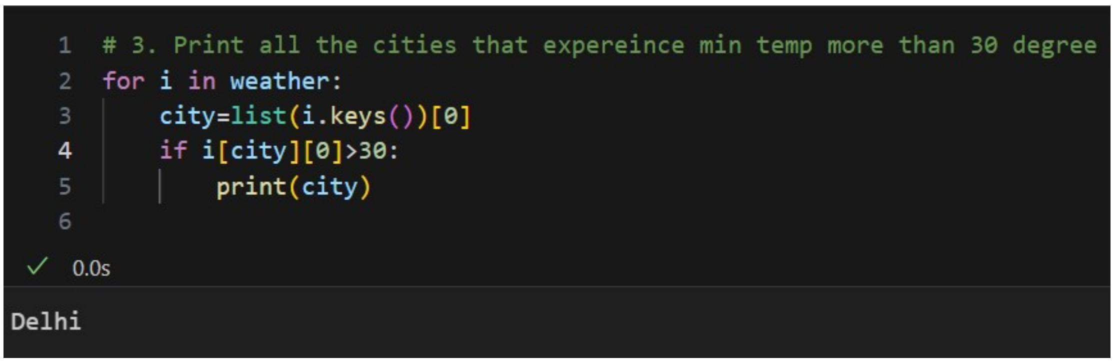


```python
# 2. Print the city with maximum/min temp
max_temp=-float('inf')
min_temp=float('inf')
max_city=None
min_city=None
for k in weather:
    city=list(k.keys())[0]
    current_min=k[city][0]
    current_max=k[city][1]
    if current_max>max_temp:
        max_temp=current_max
        max_city=city
    if current_min<min_temp:
        min_temp=current_min
        min_city=city

print(f'{max_city} with {max_temp}')
print(f'{min_city} with {min_temp}')

```

✓ 0.0s

```
Delhi with 41
Srinagar with 5
```

3. Print all the cities that expereince min temp more than 30 degree

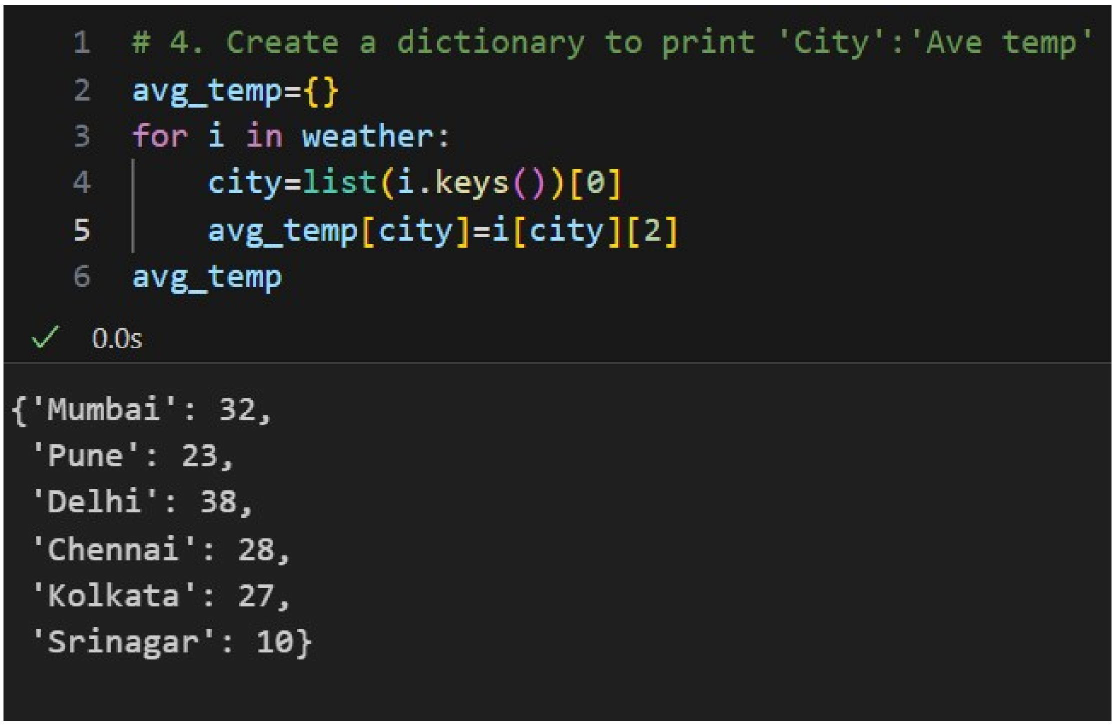


```python
# 3. Print all the cities that expereince min temp more than 30 degree
for i in weather:
    city=list(i.keys())[0]
    if i[city][0]>30:
        print(city)

```

✓ 0.0s

```
Delhi
```

4. Create a dictionary to print 'City':'Ave temp'

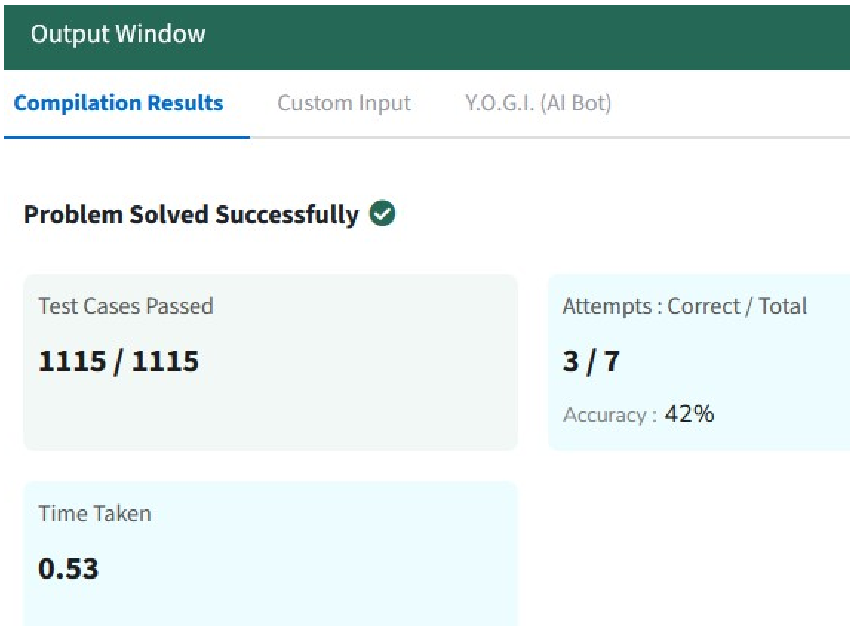


```python
# 4. Create a dictionary to print 'City':'Ave temp'
avg_temp={}
for i in weather:
    city=list(i.keys())[0]
    avg_temp[city]=i[city][2]
avg_temp
```

✓ 0.0s

```
{'Mumbai': 32,
 'Pune': 23,
 'Delhi': 38,
 'Chennai': 28,
 'Kolkata': 27,
 'Srinagar': 10}
```

Rotate Array


Output Window

Compilation Results   Custom Input   Y.O.G.I. (AI Bot)

Problem Solved Successfully

Test Cases Passed

1115 / 1115

Attempts : Correct / Total

3 / 7

Accuracy : 42%

Time Taken

0.53