Date: 3/9/25

Task - 5.1

Implement various searching and Sorting Operations

AIM:- To implement various searching and sorting operations in python programming.

ALGORITHM:-

1. Input Definition.
2. Define the function find_Employee_by_id that takes two parameters.
3. Iterate through the list: use a for loop to iterate through each dictionary in the Employees list.
4. Check for matching ID.
5. Return matching Record: If a match is found, return the current dictionary.
6. Handle No match:
   If the loop completes without finding a match, return None.

PROGRAM:-

```
def Find_employee_by_id ( Employees , target_id):
    for employee in Employees :
        if Employee ['id'] == target_id :
            return Employee
    return None.
Employee =[
{'id': 1,('name') : 'Alice', 'department':
    'HR'}
{'id': 2, 'name': 'Bob', 'department': Engineering'}.
{'id': 3, 'name' : 'charlie', 'department': 'Sales'}]
Print ( Find_ Employee_ by_id (Employees, 2))
```

OUTPUT :

{'id': 2, 'name': 'Bob', 'dept' :

Engineering }

Date: 3/9/25

Task -5.2

AIM :- To develop a python Program that sorts students records by scores in ascending order using the bubble sort algorithm.

ALGORITHM :-

1. Initialization
   - find the length of the student list → n.
2. Outer loop (passes)
   - Repeat for i=0 to n-1.
3. Track Swaps
   - set swapped = false at the start of each pass.
4. Inner loop (comparision)
   - For each j=0 to n-i-2:
     - Compare students[j]['score'] and students[j+1]['score']
     - if students[j]['score'] > students[j+1]['score']:
       - Swap them
       - Set swapped = True.
5. Early Termination
   - After the inner loop, if swapped == false, break.
6. Completion
   - The list is now sorted in ascending order of scores.

OUTPUT :-

{'name': 'Charlie', 'score': 75}

{'name': 'Diana', 'score': 88}

{'name': 'Alice', 'score': 85}

PROGRAM :-

```
def bubble_sort_scores(students):
    n = len(students)
    for i in range(n):
    swapped = False
    for j in range [0, n-i-1]:
    if students[j]['score'] > students[j+1]['score']
    students[j], students[j+1] = students[j+1],
                                 students[j]
        swapped = True.
    if not swapped:
        break
    Student = [
{'name': 'Alice', 'score': 88},
{'name': 'Bob', 'score': 95},
{'name': 'Charlie', 'score': 75},
{'name': 'Diana', 'score': 85}.
]
    Print ("Before sorting':)
    For student in students:
        Print (Student)
    bubble_sort_scores (students)
    Print ("\n After Sorting:').
For Student in Students:
    Print (Student)
```

VEL TECH - CSE	
EX.NO.	5
PERFORMANCE (5)	5
RESULT AND ANALYSIS (3)	5
VIVA VOCE (3)	5
RECORD (4)	
TOTAL (15)	
SIGN WITH DATE	15/7

RESULT :- Thus the program for various searching and sorting operations is Executed and verified Successfully.