TASK : 5.1 3/9/25

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 t
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 for matching Record: If a match is found, return the current dictionary.
6. Handle No match.

If the loop completes without find a match return name.

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', 'deparisd:}
print find-employee-by-id (employee.2))
```

Result: Thus to implement various Searching and Sorting operations in Python programming.

Output :

{'id': 2, 'name': 'Bob', 'department': 'Engineering'}

Aim: To develop a python program that starts Students records by Sources 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 at each pairs.
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'] > Student[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 to Scores.

program:

```
def bubble_sort_scares(students):
    n = len(students)
    For i in range(n):
    Swapped = false
```

Output :-

Before Sorting

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

{'name': 'Bob', 'score': 95}

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

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

After Sorting

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

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

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

{'name': 'Bob', 'Score': 95}

```
    for j in range(0, n-i, -j):
    if students[j]['score'] > students[j+1]['score']
    students[j], students[j+1] = students[j+1],
            swapped = True
            if not swapped
                break
    {'name': 'Alice', 'score': 88};
    {'name': 'Bob', 'score': 95};
    {'name': 'Charlie', 'score': 75};
    {'name': 'Dave', 'score': 85}
    ]
    print("Before sorting:")
    for student in student:
        print(student)
    bubble-sort-scores(students)
    print("\n After Sorting:")
    for student in students:
        print(student)
```

VEL TECH	
EX No.	5
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (3)	5
RECORD (4)	
TOTAL (15)	
SIGN WITH DATE	15

Result: Thus the program for various Search and Sorting operations is executed and verified successfully.