TASK 5:- Implement Various Searching and Sorting operations
Date: 3/9/25 in Python Programming

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.
a. A list of dictionaries (employees) where each dictionary represented an employee record with keys id, name, and department.
b. An integer (target_id) representing the employee ID to be searched.
3. Create through the list:-
Use a for loop to iterate through each dictionary in the employee list.
4. Check for matching ID
within the loop, check if the id field of the current dictionary matches the target_id.
5. Return matching Record.
if 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
# Test the function.
employees = [
    {id': 'name': 'Alice', 'department': HR'}
    {'id': 2, 'name': 'Bob', 'department'. Engineering'};
    {'id': 3, 'name': 'charlie', 'department': 'seales'}
]
printf (find_employee_by_id (employees,2)) #output: {'id': 2, 'name':
'Bob', 'department': Engineering'}
```

Output!

```
['id' : 2, 'name' : 'bob', 'department' : 'engineering']
```

5.2 you are developing a grade management system for a School. The system maintain a list of Student records, where each record is respected as dictionary contaiining a student's name and Score. The school needs to generate a report that displays student's scores in ascending order. you task is to implement a feature that sorts the student records by their scores using bubble sort algorithm.

Algorithm:-

1. Initialization
· Get the length of the student list and store it inn.
2. outer loop.
· Iterate from i=0 to n-1. this loop represented the number of passes through the list.
3. Track swaps
· Initialize a boolean variable swapped to false. This variable will track of any swap are made in the Corrent pass.
4. Inner loop.
· Interate from j=0 to n-i-2 (inclusive). This loop compares adjacent elements in the list and performs swap if necessary
5. Compare and swap.
for each pair of adjacent elements.
  · compare their score values.
· if Student[j]['score'] > students[j+1]['score'], swap the two elements
· Set Swapped to True to indicate that a swap was made
6. Early Termination
· After each pass of inner loop, check if swapped is false. if no swaps were made during the pass, the list is already sorted, and you can break oot pof the outer loop early.
7. completion.
· The function modifies the student list in place, storting it by score.

```
def bubble_sort_scores (students):
 n= len(students)
 for i in range (n):
# Track if any swap is made in this pass
   swapped = false
   for j in range (0,n-i-1);
    if Students[j]['scoore']> Student[is j+1]['score'];
#swap if the scores of the corrent student is greater that
the next.
Students[i], Students[i+1]= Students[j+1] [student[i]].
Swapped= True.
# if no two elements were swapped, the list is already sorted
  it not scoapped.
  break
# Example usage
Students =[
 {'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 ("In After Sorting:")
for student in Students:
   Print (student)
```

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

[illegible]

Result: Thus, the program for various searching and sorting operations is executed and verified successfully.

Output:

```
Before Sorting
{'name': 'alice', 'score': 88}
{'name': 'bob', 'score': 95}
{'name': 'charile', 'score': 75}
{'name': 'diana', 'score': 85}
After Sorting
[{'name': 'alice', 'score': 88},
  {'name': 'bob', 'score': 95},
  {'name': 'charile', 'score': 75},
  {'name': 'diana', 'score': 85}]
```