3-9-25

Task-5 implement Various Searching and Sorting operations in Python Programming.

Aim: To implement Various Searching and sorting operation in Python Programming.

Algorithm:

```
1. input definition
2. Define the function find-employee-by-id that take two parameter
   a) An integer (target_id) representing the employee ID to be
      Searched.
3. Iterate through the list
4. Return Matching Record
      if a match is found, return the current dictionary
5. Handle No Match:
      if the loop completes without finding a match, return None
```

Program 5.1

```
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':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', 'department': Engineering'}
```

output

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

Output:

```
{'name' : 'Alice', 'score': 88}
{'name': 'Bob', 'score': 95}
{'name': 'charlie', 'score': 75}
{'name': 'Diana', 'score': 85}
```

Task 5.2

Aim: To implement a feature that sorts the student record by there scores using the Bubble sort.

Algorithm:

1. Initilization
2. outer loop
3. Track swaps
4. Inner loop
5. Compare and swap
6. Early Termination
7. Completion

Program 5.2

```
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+1):
        if students[j]['score'] > students[j+1]['score']
         #swap if the score of the current student is greater
               than the next
     students[j], students[j+1] = students[j+1], students[j]
        swapped = True
        break
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("\n After sorting:")
for student in students:
    Print(student)
```

Result: thus the Program for Various searching and sorting operations is executed and verified succesfully.

VEL TECH	5
EX NO.	5
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	
RECORD (5)	
TOTAL (20)	15
SIGN WITH DATE	