B-4-a

Task 8 Implement various searching and sorting operations in python program

Aim: To implement various searching and sorting operations in python programing

Algorithm

1. Input Definition
2. Define the function find_employee_by_id
   a) a list of Dictionaries where each Dictionary represents as employee
3. Iterate through the list
4. check for matching ID:
   within the loop to iterate through each Dictionary in the employees
5. Return matching Record:

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

Restart: C:/Users/91979/Desktop/point1/b1.py

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

5.2 : you ar Developing a grade management system for a school. The system maintains a list of student records, where each record is represented as a Dictionary

Algorithm:

1. Initialization
   * get the length of the students list and store

2. outer loop:
   Iterate from i = 0 to n-1

3. Track swaps:
   * Initialize a boolean variable swapped to False

4. Inner loop:
   * iterate j = 0 to n-i-2.

5. compare and sway
   If students[j]['score'] > students[j+1][score], swap the two elements

6. Early Termination

7. completion
   * The function modifies the students list in place, sorting in by score

Program

```
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);
```

Out Put:

```
Restart: C:/Users/91979/Desktop/Print1/
Before sorting:
{'name': 'Alice', 'score': 88}
{'name': 'Bob', 'score': 95}
{'name': 'charlie', 'scare': 75}
{'name': 'Diana', 'score': 85}.
```

```
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
# If no two elements were swapped, the list is already
 sorted
 If not swapped:
    break
#example usage
    students = [
 {'name': 'Alice', 'score': 88},
 {'name': 'Bob', 'score': 95},
 {'name': 'charlie', 'score': 78},
 {'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)
```

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

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