Task-5:- Implement various searching and sorting operations in python programming. Aims To implement various searching and sorting operations in Python programming 5.1: A Company stores employee records in a list of dictionaries, where each dictionary contain id, name, and department write a function find - emp loyee-by-id that takes this list and a target emplo yee 10 as arguments and returns the dictionary of the employee with the Matching 10, or None if no such employee is found! Algorithm: 2 Define the function find-employee-by-id that a. A list of dictionaries (employees), where each diction takes two parameters: -nary represents an employee record with keysid, b. An integer (target-id) representing the employee 10 to be searched. use a for loop to iterate through each dictionarying the employees list. within the loop, check if the id field of the comen 4. Check for Matching 10: + dictionary matches-the target-id. 5. Return Matching Record: If a Match is-found, return the current dictionary 6. Handle No Match: If the 100P completees without finding a Match, return None.

primorporpodiver a 2007 Postra Las printsross business the sansignie of Primmerporgraphy 9 01 200 H- 19150 of it spread a stone a second a in misters promotes is done enough controlly conten d nome on deportment write a function find - cons eyee-by-id that takes this list and a torget emplo In as a guments and returns the dictionary output!

{'id':2, 'name': bob', 'de partment': 'engineering'} - conthiropla Chall-Inthab tugal Doct the fore than And - Implyee - by-id that JA 115t of dictionaries (compages) where each dictio more represents on employee record with keysid of sampleder (fordet-jet) repressionting the employee 10 the rate through the list: through each dictionary's remos entro biento la entro de la entro de la entro come de la entro come de la entro come de la entro come de la entro della entro de la bi tise not is At-2 set show proposition Appropriet stop to some soft order to the party of delands of the LANDER DECISIONAL HUDAREN CONTRA

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=[S'id': 1, 'name': 'Alice', 'department': 'HR'3, {'id': 2, 'name': 'Bob', 'department': en gineering's, ¿'id': 3, 'name': 'charlie', department': sales's, Print (find _employee - by - id (employees, 2)) #output: [id: 2, 'name': 'Bob', 'department': 'engineering'y seggi sonor! seggi sonor! Lie will broke supply apprillation supply apprillation of the Ethors's Ethorlis's sore's 753 Ethorlis ale Ham beginson Eligible and it among Les adjaces (88831 20021 (30)18' : 'smon's 1 PER " Special " dos" " sement?

5.2) you are developing a grade Management system for aschool. The system maintains a list of stude -ntes name and-scare-The records, where each record is represented as a dictionary combining a student's name and score. The school needs to generate a report that displays students scores in ascending order. Your task is to imple ment a feature that scorts the student records by their scores using the Bubble Sort algorithms.

Algorithm:

1. Initialization: Get the length of the students list and storeit

inn. Iterate From i=0 ton-1 (inclusive). This loop represents 2.outerloop: the number of passes through the list.

3. track swaps: initialize a boolean variable swapped to fase. This variable will track if any swaps are made in the cerrent Pass.

· Iterate From i= 0 to n-i-2 (inclusive). This loop com 4. Inner Loop: Pares adjacent elements in the list and performs swaps if necessary.

5. compare and swap:

· for each pair of adjacent elements (i.e. students[i] and students [i+1]:

· compare their score values

· if students[i]['score]>students[i+i)['score'], swapthe two

· set suapped to True to indicate that a swap was mode.

6. Carry Termination: · After each pass of the inner loop, check its wapped istalse. If no swaps were made during the Pass, the list is already sorted, and you can break out of the outer loop early.

L'SHA!! Fasminogsb' (solla'; sman' L!bill Output; E'name': 'Alice', 'score': 889 Before sorting: {'name': 'Bob', 'score': 954 l'name: charlie, score: 753 {'name': Diana', 'score': 854 After Sorting: E'name': 'charlie', 'score': 75 y {'name': 'Diana', 'score': 853 l'name': 'Alice', 'score': 883 { 'name': 'Bob', 'score': 954

101151137 SOID 18

1-29990190

```
def bubble-sort-scores (students):
        n = len (students)
        #Track ifany swap is made in this paiss
        Swapped = false
        torjin range (o, n-i-1);
          if student[i]['score']>student[i+i]('score'];
            #swap if the score of current student is gre
            ter than the next
           Students [i], students [i+i] = students [i+i]
            Students [i]
            Swapped = true
         #If no two elements were swapped, the list
          is already sorted if not swapped:
                        mes adjacent elemen
             break
                            MOPS IN THE MERCESSON
      # Example usage
                       K compose and sudp.
  Endents = [some stassorbonto ring done no
        E'name!: 'Alice', 'score!: 889,
        [hame': '80B', 's core': 953,
E'name: 'charlie', 'score': 753,
  l'name': 'oiana', 'scoré: 854
Print ("Before sorting:")
forstudent in students:
print(student)
                                 Manys Natha
                             DINE ON TE BEIDING
     bubble_sort_scores (estents)
     Print ("In After Sorting:")
    for Studentin students:
       Print (student)
```

and sorting operation Brint ("In After. Res vitin 150s print ("Betweent in so Print student students Sorting! he Program tor v PERFORMANCE (5) RESULT AND ANALYSIS 的fac(图) RECORD (5) VIVA VOCE (5) SIGNWITH 1erifie

Success fully