Task. . Implement various Searching and Sorting operations in pythen programing

Aim:

operations in python progressing.

Ed. A company stores employee records in a list dictionaryies, where each dictionary contains. id., name and department. Write a function find-employee byid that takes this list and a target employee 1D as arguments and returns the dictionary of the employee with the matching 1D, or None if no such employee is found.

Algorithm:

- 1. Input Definition:
- 2. Define the fortion find-employee-by-id that takes two parameters:
 - · a. A list of distionary (employees), where each dictionary represents an employee record with key id, name, and department.
 - b. An integer (target-id) representing the employee ID to be sourched.
- 3. Herate Trough the list: Use a for loop to iterate through each dictionary in the employees list.
- h. Check for Metching ID: within the loop, check if the id field of the current dichbrary matches the target id.
- 5. Return Macthing Record: If a motten is found, return the current dictionary
- 6. Handle No match: If the loop completes without finding a match return None.

Output: { 'id': 2, 'name! : Rob', 'department': Engancening'? as a factorial of principles and part and invidence at a more apolitica parte programa la la a grandfall down arader gargermentally The regions with a little of their etransferred been man are tyrade a bas full side solde tall to received alle conter lovo attentopes co to with the redding . The sopplant Liver I spelyme in : mitting profinded togeth a patient the thinking that entire you a. A list at dictioning (employed) where distingly seprends on employed formation May if name and department il marini (bi taprat) reportini na me dust it was a set of the list of art forth of the photosopher with in the moragens his Charle of modeling this within the born back earstone promote theorem at to bird his at the Maker Mach of Johns to a laced griffson unital Hondle No modely It the loop completes without and book a water a part of

```
Program:
def find-employee-by-id (employees, target-id):
 for employee in employees:
    if employedid) == target_id:
       retorn employee
retorn None
# Test the Ronation
employee = [
 S'id: 'name': 'Alice', 'depter trant': 'HR'3,
{ lide !: 2 'name !: Bob', 'deportment': Engineering' };
{'id': 3,! nome! Charlie', 'department!: 'Sales'}.
print (find_employee_by-id (employee, 2)) # output:
[id: 21, name : 130b; department: ! Engineering }
```

system for a school. The system maintains a list of statent records, where each record is represented as a dictionary containing a statent's name and score. Your task is to implement a fastered that sort the statent record by their scores using the Bubble sort algorithm.

Algorithm:

- 1. Initialization.
 - · Get the length of the students like and store it in n.

strang's month

(38: growt / smail farmer)

- 2. Outer Loop.
 - · I terate from i=0 to not (inclusive). This loop represents the number of passes through the list
- 3. Torack Suaps:
 - · Initialize a boolean variable swapped to false. This variable will track if any swaps are made in the current posses.
- h Inner loop:
 - Iterate from j=0 to n+i-2 (inclusive). This bop cormposes adjacent elements in the list and performs supps if necessary.
- 5. Compare and Swap:
 - · For each poir of adjacent element (i.e. student [i] and students [i+1]):
 - · compare theirs score values
 - o If stooms Iil (score), stodents [i+1]['score'], scop the two elements.
 - o set supped to True to indicate that a swap was made.

Output: Before sorting: { name! : 'Alice!; 'score!: 88} {'name': 'Bb', 'sore': 95} { name !: 'Charlie', 'score! 75] {'name': Diana; sore! 85) Smith of april inthositotial After sorting: {'name': 1 charlie; sorce = 75] { name ! Diane 1, ! scare !: 85} {!none!: Alice!, 150061: 88} E'name's 'Bob', Isome': 95] Initialize a backer viviable rupped to Mess pris Is wast the aldonor sitt allo mede in the court result invote from joods you a (industrial) This bear is improved adjacent character in the test as traducts of transfer transfer to view does of ((1+1) strabute boxo (6) Compare theirs since values Carol Theil states of must still statute II. swap the two clanerts. a but shouth it of suff of baggious to so

6. Early Termination:

· After each pass of the inner loop, chock if supposed is false. If no suarps were made during the pass, the list is already sorted, and you can break out of the orater loop early.

7. Completion:

. The function modifies the soldents list in place, sating it by score.

Program:

def bubble-sort-score (student):

n=len (stodents)

for i in range (n):

Track if any swap is made in this pass swapped= False

for jin rouge (o.m.i-1):

if students [i)['score'] > students [i+1]['score']:

Swap if the store of the ownert statent is greater than the next.

Students [i], student[i+1]=students [i+1], student[i] Shapped = True

If no two elements were swapped, the list is decong sorted.

if not swapped:

break:

Example usage

Student = [

]

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

{ 'rome': 'Bob', 'score'; 95),

{'rand: 'kharlie', 'score'; 75},

{'namé: 'Diana', Iscare!: 853

print ("Before sorting:") for student in a students: print (student) bubble_sort_score (students) print ("In A fler sorting: ") for stodent in stodents: (tradute) tring

VELTECH	
EX No.	
VIVA VOCE (3)	
STAL (15)	
SIGNALL	

Result:-Thus, the program for various Scarching and sorting grenation is executed and verified successfully