Task 5- Implement various, searching and soxfing operations in Python Programming. Aim: To implement various searching and sorting openations in Python Programming. Stores employee records in a Cit of dictionaries, where each dictionary contains id, name and department. write a function find employee by id that takes this list and a taxget employee ID as arguments and returns the dictionary of employee with the matching ID , or None if no such employee is found. A gorthm.

1. Input Definition:

2. Define the function find-employee-by-id that takes two Parameters; tionaries, whore each dictionary represents an a) A list of dictionaries, where each dictionary represents and defortment.

a) A list of record with keys id, name, and defortment.

employee representing employee 1D to be searched.

I tenate through the list:

Use a few loop to itenate through each dictionary in the 4. check for matching 1D:

within the loop, check if the id field of the current within the loop, check it the target id.

dictionary matches the target id. Refurn matching Record; return the current dictionary of a match, is found, return the current dictionary 5. Return Hardle No match:

To the loop completes without finding a match; 6. Hardle No match: Point (student) Charlent return None. (1: Enit 102 1002 - 1002 . 9 double (1: Enit 102: ) mediable at thekute it 19. et Mil got de l'Original a la forma la forma la forma de l'anti-F: 'srore' , sileado' : aron'! \$ 18 5 8600 ; Boild : 9000 7 orone : Mara . Store : 30 fame : Bob 'score ags (88: 'SKOLZ' , 'DINA' : SMALL for more silvati . smant rane : Sib Store : 95 Per: sunz' comia : sans

Rogram: det find - employee - by - id (employees, target - id). for employee in employees!

if employee [id'] = = target\_id: return employee

return None

Test the function

employees = [...

Lid: 1, 'name': 'Alice', department': 'AR' 4 i 'id': 2; name : 'Bob', 'defartment': Engineoung } i id : 3, 'name'; charlie', department : 'sales' Print ( Rind\_employee\_by\_id (employees, 2)) # output . fid ; 'name': 'Bob', de Portment': Engineering's steptio == = Restout: c: /users / 91979 | Desktop / Print 1/3. fg. f'id': 2, 'name'; Bob', department'; Engineering's various data riggs, list Tuple and Distinguy in Mules Programming was used and voished successfully · cust Hon

Program: 18 32 too gold season was easy trams from def bubble - sort - scores (students). n= len (students) n=len (students)

for i in range (n):

# Track if any swap is made in this Paus

swapped = False

for i in range (o, n-i-i): if students (i] ('score') > students (i+)['score']: It swap if the some of consent student is greater than to next students [i], students [i+i] = students [i+i], students [i] next students [i], students [i+] = Swapped = True

If no two elements were swapped, the lit is abandy some

if not swapped:

break

treak

t students = C s'name': Alice', 'score': 884, 2 'name': 'charlie', 'score'; 753, at anoton ponoton Ename : Diana , Score ; 853 of match is feund prived ("Before sorting:") e tiville Normatch. For students in students: of the loop completes bubble sort scores (students) Print ("In after sorting: ") return work for student in students: Print (student) outPut: == Restord: c: lusery 191979 | Desktop | Private | ts.P. after sorting: Defore sorting! i'name : 'charlie', 'score' : 453 finame': 'Alice', Score: 88 } {'name'; Diana', 'score'; 85] I name : 'Bob', Score : 954 {'name': 'Alice', 'score': 98} finame : 'charlie', 'score': 75} frame': 'Bb', 'score': 955 l'name: 'Diana', 'score : 859

Scarching and sosting operation in

5.2: You are developing a Grade management system for a school. The system maintains a list of student records, where each record is depresented as a dictionary containing a students name and score. The school need to generate a report that displays students scores in ascending order. Your task is to implement a feature that sorts the student records by their scores using Bubble-sort Algorithm.

Aborithm:

Toi fializations To falization:
The students list and store it in no store to no store it in no store toop: Track swaps: bookan variable swapped to Falk.

\* Initialize or bookan variable swapped to Falk. This variable will track if any scoops are made in ensemble This variable will

Timer loop

Timer loop

This loop compare adjacent

This loop compare adjacent

The list and performs swaps if necessary

Elements in the list and performs

The compare and swap

The adjacent elements

The for each pair of adjacent elements

The score values 5. compare and swap: > compare their score values students [Iti] ['Score'], swap the if students [I] ['Score'] > students [Iti] ['Score'] two elements.

True to indicate that a swap set swapped to True to indicate that a swap made 6. Early Termination:

\* After each part of the inner look check if swapped is falle. If no swaps were made during part. 4. completion. PERFORMANIA MAINS (S) S TOTAL (20) SIGN WITH DATE Thus, the program for vocious searching and sorting Peration is executed and verified successfully.