Task - 5

Date 03-9-25 IMPLEMENT VARIOUS SEARCHING AND SORTING OPERATIONS IN PYTHON

Aim :- To implement various searching and sorting operat-ons in python programming.

Algorithm :-

→ Input Definition:

→ Define the function find-employee-by-id that takes t

Parameters:

a) A list of dictionaries (employees), where each dictionary represents an employee record with keys id, name and department.

b) An integer (target-id) representing the employee ID to be searched.

→ Iterate Through the list:

use a ~~foor~~ for loop to iterate through each dictionary in the employee list

check for matching ID:

within the loop, check if the id field of the current dictionary matches the target_id

> Return Matching Record:

If a match is found, return the current dictionary.

> Handle No match:

If the loop completes without finding a match, return None.

```
Output : {'id': 2, 'name': 'bob', 'department':
                                   'engineering'}
```

Program

```
def find-employee-by-id (employees, target-id);
    for employee in employees:
        if employee ~~in~~ em ['id']==target-id ;
        return employee
    return None
# Test the-function
employees = [
{'id' : 1,'name! 'Alice', 'department':'HR'},
{'id'!2, 'name' : 'Bob', 'department': 'Engineering'},
{'id', 'name' : 'charlie', 'department', 'sales'},
]
Print (find-employee-by-id(employees, 2))# output,
                    {'id' : 2,'name':'Bob','department':
                                            'Engineering'}
```

Task-5.2

You are developing a grade management system for a school. The system maintains a list of student records, where each record is represented as a dictionary containing a student's name and score. The school needs to generate a report that displays student's score in ascending order. Your task is to implement a feature that sorts the student records by their scores using the bubble sort algorithm

Algorithm:-

1. Initialization:

->Get the length of the students list and store it in n.

->Outer loop:

->Iterate from i=0 to n-1 (inclusive). This loop represents th number of passes through the list.

-> Track swaps:-

-> Initialize a boolean variable swapped to false. Th variable will track if any swaps are made in the curren pass.

Inner loop:-

Iterate from j=0 to n+2 (inclusive). This loop compar adjacent elements in the list and performs swaps if necessary.

compare and swap:

->Iterate from j=0 to n-i-2 (inclusive). This loop comp adjacent elements in the list and performs swaps if necessary.

->for each pair of adjacent elements (i.e., students[j and students[j+

->compare their score values.

->If students [j]['score'] > students [j+1]['score'], swap two elements.

-> set swapped to true to indicate that a swap w made.

output:-

Before sorting:

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

After sorting :

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

6. Early termination

-> After each pass of the inner loop, check if swapped is false. If no swaps were made during the pass, the list is already sorted, and you can break out of the outer loop early.

7. Completion :-

-> The function modifies the students list in place, sorting it 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):
          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[i]
              swapped = ~~false~~ True
      ~~for j in range (0,n-i-1)~~
      # 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': 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)
```

VEL TECH - CSE	
EX NO.	5
PERFORMANCE (5)	5
RESULT AND ANALYSIS (3)	5
VIVA VOCE (3)	5
RECORD (4)	
TOTAL (15)	
SIGN WITH DATE	15

Result :- Thus, the program for various searching an sorting operations is executed and verified successful