Task -5 : Implement various searching and sorting operations in python programming.

1/9/25

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

Algorithm:

1. Input Definition:
2. Define the function find_employee_by_id that takes two parameters;
3. Iterate through the list
4. use a loop to iterate through each dictionary in employee list.
5. Return matching found, return the current dictionary.
6. Handle No match.
   ~~If~~ the loop completes without finding match, return None.

Program 5.1

```
def find_employee_by_id(employees, target_id);
for employee in employees:
    If employee[id] ==target_id:
        return employee.
    return None
# Test function
employee =[
    {'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': 'Bob', 'department': 'engineering'}
```

output:

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

output 1.

```
Before sorting:
{ 'name': 'Alice', 'score': 88}
{ 'name': 'Bob', 'score': 95}
{ 'name': 'charile', 'score': 75}
{ 'name': 'Diama', 'score': 85}
After sorting:
{ 'name': 'charlie', 'score': 75}
{ 'name': 'Diane', 'score': 85}
{ 'name': 'Alice', 'score': 88}
{ 'name': 'Bob', 'score': 95}
```

1/4/25
5.2 :- Algorithm

Aim: To implement a feature that sorts the student records by their scores using the Bubble Sorth Algorithm.

Algorithm:

1. Initialization:
   • Get the length of students and store it in.
2. outer loop;
   • Iterate from i=0 to n-1. This loop represents the number of passes through the list.
3. Track swaps:
   • Initilize boolean variable swapped to false, this variable will track if any swaps are made in current pass
4. Inner loop:
   • Iterate from j=0 to n-i-2 (inclusive). This loop compress adjacent elements in list and performs swaps if necessary.
5. Early Termination:
   After each pass of inner loop check if swapped is flase. If no swaps were during pass, the list is already sorted and you can break
6. Completion;
   • The function modifies the students list in place, sorting it by score.

Program;

```
def bubble_sort_scores(students):
    n= len(students)
    for 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]['~~sc~~ score'];
     # swap if score of current student is greater than next
    students [j], student [j+1] = student [j+1], student [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': 75},
    {'name': Diane; score': 85},
 ]
   print ("Before sorting.")
   for student in students :
       Print(student)
   bubble_sort_scores(students)
   print ("\n After sorting :")
   for students in students:
       print (student)
```

Result: Thus the program for various searching and sorting operation is executed and verified successfully.

VEL TECH - CSE	
EX NO.	5
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	
TOTAL (20)	15
SIGN WITH DATE	