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

Aim: 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 contains id, name & dept.

Algorithm:

1. Input Definition
2. Define the function find_employee_by_id that takes two parameters:
   a. A list of dictionaries (Employees).
   b. An integer Representing the Employee ID to be searched.
3. Iterate Through the list;
4. Check for Matching ID;
5. Return Matching Record;
6. Handle No match.

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 = [
  {'id': 1, 'name': 'Alex', 'department': 'HR'},
  {'id': 2, 'name': 'Bob', 'dept': 'Engineering'},
  {'id': 3, 'name': 'Charlie', 'department': 'Sales'},
]
print ('find_Employee_by_id (Employees, 2)) #output ((
{'id', 2, 'names', 'Bob', 'dept', 'Engineering'}
```

Dstart : C : Users / 81879 / Desktop

('id', : 2, 'name', 'Bob', 'dept' : 'Engineering'

5.2.

Aim; 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 & score.

Algorithm:

1. Initialization:
   - Get the length of the students list and store it as n.
2. Outer loop:
   - Iterate from i=0 to n-1. This loop represents the number of passes through the list.
3. Track swaps.
4. Inner loop
5. Compare and swap
6. Early Termination
7. Completion

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'] > student [j+1]['score']:
        students[j], students[j+1] = students[j+1], students[j]
        swapped = True
    if not swapped:
        Break
students = [
    {'name': 'Alice', 'score': 88},
    {'name': 'Bob', 'score': 95},
    {'name': 'Diana', 'score': 85}
]
print("Before sorting:")
for student in students:
    print(student)
```

Restart : C: /users/91920 /Desktop /print):

Before sorting

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

{'name': 'Bob', 'score': 98}

{'name': 'Charlie', 'score': 75}

{'name': 'Dana', 'score': 85}

```
bubble_sort_scores(students)
print("\n After sorting:")
for student in students:
    print(student)
```

VEL TECH	
EX NO.	5
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	
TOTAL (20)	
SIGNATURE DATE	

Result: Thus the program for implementing various searching and sorting operations in python programming executed successfully.