AI Assisted Coding - Lab Test 2

Htno : 2503A52L10

Subgroup M

M.1 — [S15M1] Stable sort employees by dept asc, salary desc

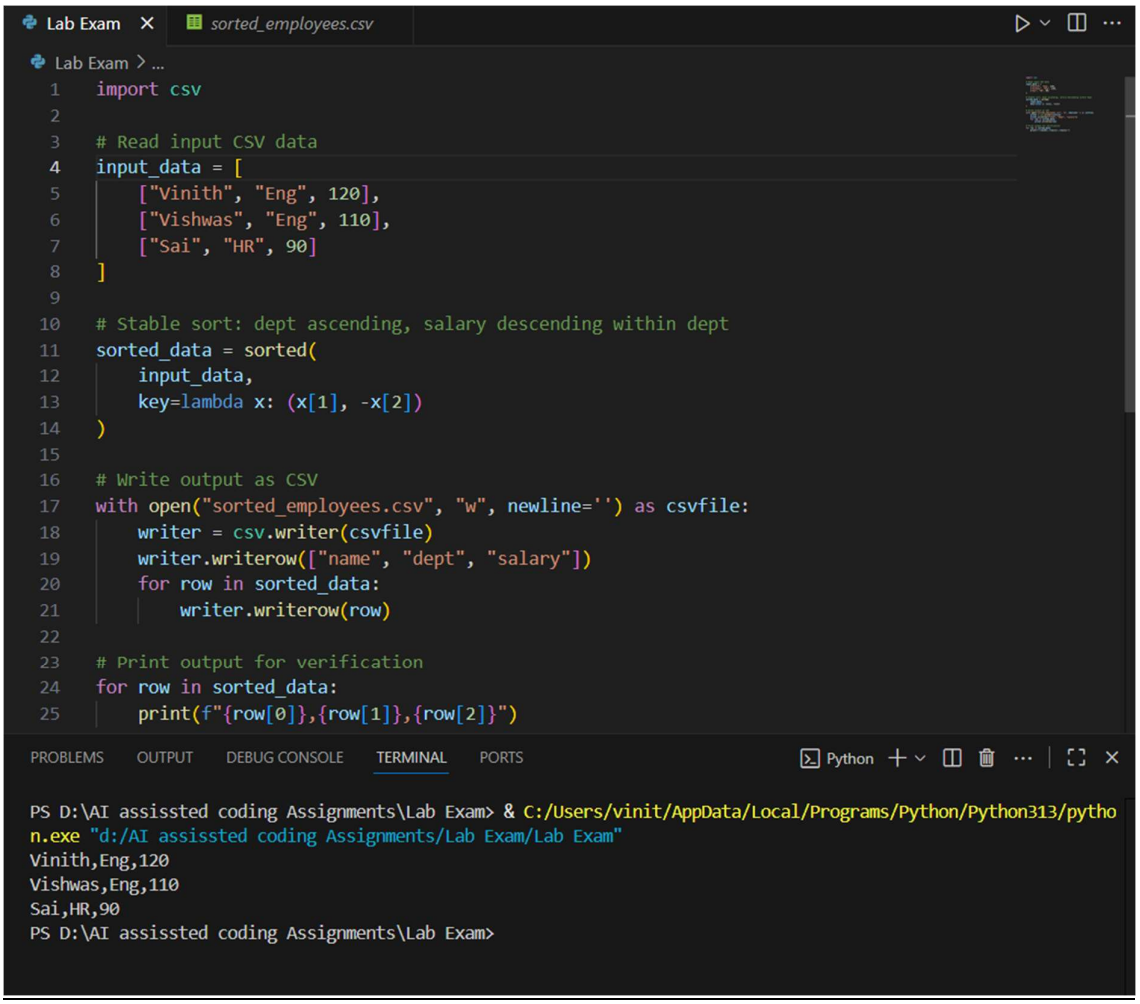
Prompt :

Write a python program that takes a list of employee records(each with Name, Department, and Salary) then sort them department in ascending order and salary in descending order within each department then display the output in separate CSV file.

Code:

```
import csv
input_data = [
    ["Vinith", "Eng", 120],
    ["Vishwas", "Eng", 110],
    ["Sai", "HR", 90]
]
sorted_data = sorted(
    input_data,
    key=lambda x: (x[1], -x[2])
)
with open("sorted_employees.csv", "w", newline='') as csvfile:
    writer = csv.writer(csvfile)
    writer.writerow(["name", "dept", "salary"])
    for row in sorted_data:
        writer.writerow(row)
for row in sorted_data:
    print(f"{row[0]},{row[1]},{row[2]}")
```

Output :



Explanation :

I. Firstly imported CSV file
II. Then defined employee data
III. Then sorted department by ascending then salary by descending stable
IV. Then sorted_employees .csv file was generated and the output was given in that CSV file

M.2 — [S15M2] Process movement commands

Prompt :

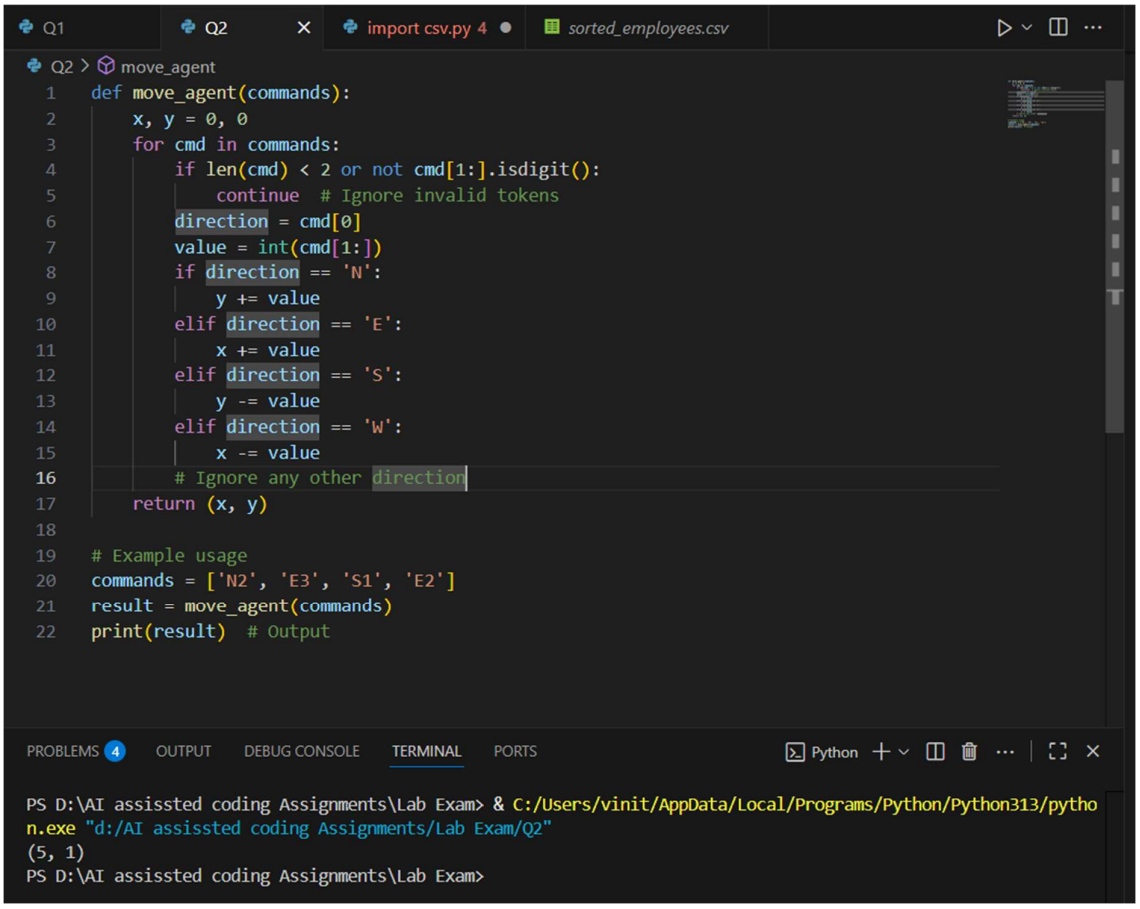
Write a python function that takes a list of movement commands for an agent starting at position(0,0) on grid. Give me related code based on related data.

Code:

```
def move_agent(commands):
    x, y = 0, 0
    for cmd in commands:
        if len(cmd) < 2 or not cmd[1:].isdigit():
            continue  # Ignore invalid tokens
        direction = cmd[0]
        value = int(cmd[1:])
        if direction == 'N':
            y += value
        elif direction == 'E':
            x += value
        elif direction == 'S':
            y -= value
        elif direction == 'W':
            x -= value
        # Ignore any other direction
    return (x, y)

# Example usage
commands = ['N2', 'E3', 'S1', 'E2']
result = move_agent(commands)
print(result)  # Output
```

Output :



Explanation :

I. The function move_agent starts the agent position(0,0)
II. It loops through each movement command in list
III. Each command is checked for validity
IV. The direction determines which axis to change
V. Invalid commands are ignored
VI. Functions returns final position for all valid moves
VII. It generated the output for given values.