Assignment X: XXX
Programming report

s1234567 & s7654321

Algorithms and Data Structures 2020

1 Problem description

...

2 Problem analysis

...

3 Program design

...

4 Evaluation of the program

...

5 Process description

...

6 Conclusions

...

7 Appendix: program text

```python
#Nima

from data_loader.load_cities import load_cities
from data_loader.load_missiles import load_missiles
from map.graph import build_graph
from algorithms.pathfinding import shortest_path

# - Load Cities -
cities = load_cities("data/cities.json")
print(" Cities Loaded:")
for city in cities:
    print(f"   {city.name} ({city.country}) - Type: {city.city_type} -
       Defense: {city.defense}")

# - Load Missiles -
missiles = load_missiles("data/missiles.json")
print("\n Missiles Loaded:")
```

```python
for m in missiles:
    print(f"   {m.name} - Damage: {m.damage}, Stealth: {m.stealth}, Max Range: {m.max_range}")

# - Build Graph -
graph = build_graph(cities)
print("\n Graph Edges:")
for u, v, data in graph.edges(data=True):
    print(f"   {u} <--> {v} = {data['weight']:.2f}")

# - Sample Path Test -
print("\n Sample Pathfinding Test:")

start_city = "Hamedan"
target_city = "TelAviv"
sample_missile = missiles[0]  # First missile from list

path, distance = shortest_path(graph, start_city, target_city, sample_missile.max_range)

if path:
    print(f" Path found from {start_city} to {target_city}: {'     '.join(path)}")
    print(f"    Total distance: {distance:.2f} km")
    if distance > sample_missile.uncontrolled_range:
        print(f"    Missile exceeds uncontrolled range ({sample_missile.uncontrolled_range} km)     needs reprogramming")
    else:
        print(f"    Within uncontrolled range     no reprogramming needed")

    # Check defense
    target_defense = next(c.defense for c in cities if c.name == target_city)
    if sample_missile.stealth > target_defense:
        print(f"    HIT SUCCESSFUL     Damage: {sample_missile.damage}")
    else:
        print(f"    Intercepted by enemy defense (defense level: {target_defense})")
else:
    print(f" No valid path found within range ({sample_missile.max_range} km)")


# Yeeaah Buuuuudddyyyyyyyyyyyyyy
```