Assignment: Vending Machine Simulation Using DFA, NFA, and Python (20 marks)

Theory of Computation
Course ID: CSE6344

Objective:

The purpose of this assignment is to design and implement a **vending machine simulator** using the concepts of **Deterministic Finite Automata (DFA)** and **Nondeterministic Finite Automata (NFA)**. You will apply formal language concepts to represent the machine's behavior and simulate its operations using **Python**.

You are tasked with designing a vending machine that sells the following items:

Item	Price (RM)
Water	2.00
Soda	3.50
Chips	4.50
Chocolate	6.00
Coffee	7.50
Juice	8.50
Sandwich	9.00
Energy Drink	10.00

Machine Behavior:

- The machine only accepts values of **RM0.5** and **RM1**.
- **All other values (RM5, RM10, RM20, RM50, RM100)** are rejected and returned to the user.
- If the inserted amount matches the item price, the item is **dispensed**.
- If the inserted amount exceeds the price, the machine will:
  - ❖ **Dispense the item**
  - ❖ **Return the excess money as change**
- The user can select a new item after each purchase. *(This point in code implementation only)*

The vending machine's behavior can be described using formal language concepts as follows:

- **Alphabet (Σ)**:

  Σ={0.5,1, Invalid}

Where:

0.5 represents RM0.5 value, **1** represents RM1 value, **Invalid** represents values like RM5, RM10, RM20, etc.

Part 1: Design the NFA

1. **Design a Nondeterministic Finite Automaton(s) (NFA(s))** for the vending machine for all items.
2. In the NFA:

    - When deemed necessary, use epsilon transitions.

3. Provide a clear state diagram for the NFA(s).

Part 2: Convert the above NFA(s) into DFA using Subset Construction Method

1. Clearly represent the states and transitions using a state diagram and transition tables.

Part 3: Python Implementation

You are required to simulate the vending machine using **Python**.

Requirements:

1. Implement both the **DFA** and the **NFA** using Python Programming Language.
2. The program should:

    - Allow users to select an item.
    - Simulate value insertion using RM0.5 and RM1.
    - Track the cumulative amount and display the state transitions.
    - Dispense the item and calculate the change if applicable, return it via epsilon transitions in NFA.
    - Reject invalid values and return them to the user.

3. Provide appropriate outputs displaying the machine's state at each step.

Submission Guidelines

Your submission should be in the form of a well-structured **report**. The report must include the following sections:

1. **Title Page**
    - Assignment Title: **Vending Machine Simulation Using DFA, NFA, and Python**
    - Group ID
    - Your Names

- Student IDs
- Course Name and Code
- Instructor's Name
- Submission Date

2. **Table of Contents**
   - Provide a clear table of contents with proper headings and page numbers.
3. **Introduction**
   - Briefly describe the purpose of the assignment.
   - Explain the concept of vending machines and how automata theory applies to simulate their behavior.
4. **Problem Statement**
   - Clearly state the given problem, item prices, and the machine's behavior.
6. **NFA Design**
   - Present all state diagrams for the **Nondeterministic Finite Automaton (NFA)**.
5. **NFA to DFA Conversion**
   - Show the conversion process in detail using subset construction method.
7. **Python Implementation**
   - Include your Python code simulating both the DFA and NFA.
   - Provide clear comments within the code to explain its functionality.
8. **Test Cases and Results**
   - Perform multiple test cases to demonstrate the correctness of your simulation.
   - Include screenshots or output logs showing different scenarios:
     - Successful purchases
     - Change returned
     - Invalid values rejected
     - Exact payment scenarios
9. **Conclusion**
   - Summarize your learning experience.
   - Discuss any challenges faced and how you overcame them.
10. **References**
    - Cite any resources, textbooks, research papers, or online materials used in completing the assignment. *(If any)*

Evaluation Criteria for 20% Assignment

[1] **Design and Explanation** **8%**

[2] **Python Implementation** **8%**

[3] **Report Quality and Presentation** **4%**