**3-3 Project One Milestone One: Vector Data Structure**

**Pseudocode: ABCU Course Loader and Query**

**Data Types & Structures**

**STRUCT Course**

**id: String**

**name: String**

**prerequisites: List<String>**

**Sample Course Data (CSV lines):**

**CSCI100,Introduction to Computer Science**

**CSCI101,Introduction to Programming in C++,CSCI100**

**CSCI200,Data Structures,CSCI101**

**MATH201,Discrete Mathematics**

**CSCI300,Introduction to Algorithms,CSCI200,MATH201**

**CSCI301,Advanced Programming in C++,CSCI101**

**CSCI350,Operating Systems,CSCI300**

**CSCI400,Large Software Development,CSCI301,CSCI350**

**1. Load and Parse File\*\***

**FUNCTION loadCourses(filePath: String) → List<Course>**

**OPEN file at filePath**

**IF file cannot be opened THEN**

**PRINT "Error: Cannot open file"; EXIT**

**ENDIF**

**courses ← empty list of Course**

**FOR each line in file DO**

**trimmed ← trim(line)**

**IF trimmed is empty OR startsWith(trimmed, '#') THEN**

**CONTINUE**

**ENDIF**

**tokens ← split(trimmed, ',')**

**IF length(tokens) < 2 THEN**

**PRINT "Format error: '" trimmed "'"; EXIT**

**ENDIF**

**c ← new Course**

**c.id ← trim(tokens[0])**

**c.name ← trim(tokens[1])**

**c.prerequisites ← empty list**

**FOR i FROM 2 TO length(tokens)-1 DO**

**prereq ← trim(tokens[i])**

**IF prereq ≠ "" THEN**

**APPEND prereq TO c.prerequisites**

**ENDIF**

**ENDFOR**

**APPEND c TO courses**

**ENDFOR**

**CLOSE file**

**RETURN courses**

**2. Validate Prerequisites**

**FUNCTION validateCourses(courses: List<Course>) → Boolean**

**allIDs ← set of c.id FOR EACH c IN courses**

**FOR EACH c IN courses DO**

**FOR EACH prereq IN c.prerequisites DO**

**IF prereq NOT IN allIDs THEN**

**PRINT "Validation error: Prerequisite '" prereq "' for course '" c.id "' not found"**

**RETURN False**

**ENDIF**

**ENDFOR**

**ENDFOR**

**RETURN True**

**3. Search for a Course**

**FUNCTION findCourse(courseID: String, courses: List<Course>) → Course or null**

**FOR EACH c IN courses DO**

**IF c.id == courseID THEN**

**RETURN c**

**ENDIF**

**ENDFOR**

**RETURN null**

**4. Print Course and Prerequisites (Recursive)**

**PROCEDURE printCourse(courseID: String, courses: List<Course>)**

**c ← findCourse(courseID, courses)**

**IF c IS null THEN**

**RETURN // Course not found**

**ENDIF**

**PRINT c.id ": " c.name**

**IF c.prerequisites IS empty THEN**

**PRINT " [No prerequisites]"**

**ELSE**

**FOR EACH prereqID IN c.prerequisites DO**

**printCourse(prereqID, courses)**

**ENDFOR**

**ENDIF**

**5. Main Program Flow**

**FUNCTION main()**

**filePath ← prompt("Enter CSV file path (or press Enter for default):")**

**IF filePath == "" THEN**

**filePath ← "default/path/to/courses.csv"**

**ENDIF**

**courses ← loadCourses(filePath)**

**IF NOT validateCourses(courses) THEN**

**PRINT "Course list validation failed."; EXIT**

**ENDIF**

**targetID ← prompt("Enter course ID to display:")**

**printCourse(targetID, courses)**

**END FUNCTION**