Step 1 open the file

OPEN file FOR reading

IF file cannot be opened THEN

DISPLAY "file could not be opened"

EXIT

ELSE

“File opened successfully”

ENDIF

Step 2: Create a place to hold all courses

CREATE Binary Search Tree ABCU

Step 3: Read data from file, line by line

WHILE there are lines in file

  READ line

SPLIT line by commas > tokens

               END WHILE

 Step 4: Extract course information

  courseNumber = tokens[0]

  name = tokens[1]

  prerequisites = all remaining tokens (tokens[2] > tokens [n])

 Step 5: Create course Object

CREATE new Course object

Course.Number = courseNumber

Course.Name = name

Course.Prerequisites = prerequisites

Step 6: Insert Course Function

FUNCTION insertCourse(ABCU.root, Course)

INPUT: File tree, Courses

OUTPUT: Course Objects

IF (root is NULL) {

root equals new Node(Course)

RETURN root

END IF

IF (course.number < root.courseNumber)

Root.left = insertCourse(root.left, course)

ELSE IF ( courseNumber is less than root.course.courseNumber THEN

Root.right = insertCourse(root.right, course)

ELSE

DISPLAY “Duplicate course ID found: ” course.courseNumber

END IF

RETURN root

END FUNCTION

7: Store the course in the list

FOR (each token after position 1)

ADD each tokens[i] into course.prerequisites

END FOR

IF ABCU.root IS NULL THEN

SET ABCU.root = new Node (Course)

ELSE

CALL insertCourse (ABCU.root, Course)

END IF

Step 8: Search the Data Structure

FUNCTION searchCourse (ABCU.root, course)

INPUT: File Tree, Courses

OUTPUT: Course Information printed

IF (root is NULL)

Display “Course not found: “ + courseNumber

Return

END IF

IF root.course.prerequisites IS EMPTY

DISPLAY "Prerequisites: None"

ELSE

DISPLAY “Prerequisites: “

FOR EACH prerequisite in root.course.prerequisites

CALL searchCourse((ABCU.root, course)

END FOR

END IF

END FUNCTION

Step 9: Close File

                Close File for reading