**Walter Chico**

**6-2 Submit Project One**

**4/13/2025**

*Vector Pseudocode*

// Read courses from a file and save them

FUNCTION ReadCourses(fileName)

TRY to open the file with name fileName

IF file can't be opened THEN

PRINT "File not found or can't be opened."

RETURN

END IF

FOR each line in the file DO

IF the line is not written right THEN

PRINT "Bad line in file. Skipping."

CONTINUE

END IF

SPLIT the line into courseDetails (like ID, name, prerequisites)

CALL AddCourseToList(courseDetails)

END FOR

CLOSE the file

END FUNCTION

// Turn course details into a course and add to the list

FUNCTION AddCourseToList(courseDetails)

course = MakeCourse(courseDetails)

ADD course to courseList

END FUNCTION

// Make a course object from the details

FUNCTION MakeCourse(courseDetails)

CREATE new course

course.id = courseDetails[0]

course.name = courseDetails[1]

course.prerequisites = courseDetails[2]

RETURN course

END FUNCTION

// Find a course and show info

FUNCTION FindCourse(courseList, searchWord)

FOR each course in courseList DO

IF course.id is same as searchWord OR course.name has searchWord THEN

CALL ShowCourse(course)

END IF

END FOR

END FUNCTION

// Show course details

FUNCTION ShowCourse(course)

PRINT "Course ID: " + course.id

PRINT "Course Name: " + course.name

PRINT "Prerequisites: " + course.prerequisites

END FUNCTION

*Hash Table*

// Create the hash table

Make a table with 179 empty lists called bidTable

// Function to create a hash key from a bid ID

FUNCTION GetHashKey(bidID)

SET total = 0

FOR each character in bidID DO

ADD the numeric value of the character to total

END FOR

SET key = total MOD 179

RETURN key

END FUNCTION

// Function to insert a bid into the hash table

FUNCTION InsertBid(bid)

SET key = GetHashKey(bid.id)

ADD bid to bidTable at index key

END FUNCTION

// Function to print all bids

FUNCTION PrintAllBids()

FOR each list in bidTable DO

FOR each bid in the list DO

PRINT bid.id, bid.title, bid.amount, bid.fund

END FOR

END FOR

END FUNCTION

// Function to remove a bid

FUNCTION RemoveBid(bidID)

SET key = GetHashKey(bidID)

FOR each bid in bidTable[key] DO

IF bid.id is equal to bidID THEN

REMOVE bid from the list

BREAK the loop

END IF

END FOR

END FUNCTION

// Function to search for a bid

FUNCTION SearchBid(bidID)

SET key = GetHashKey(bidID)

FOR each bid in bidTable[key] DO

IF bid.id is equal to bidID THEN

RETURN the bid

END IF

END FOR

RETURN an empty bid (not found)

END FUNCTION

// Function to clear the table

FUNCTION DeleteTable()

FOR each list in bidTable DO

CLEAR the list

END FOR

END FUNCTION

*Tree Pseudocode*

// A course has a number, title, and list of prerequisites

DEFINE Course

courseNumber

courseTitle

prerequisites

END DEFINE

// Load courses from a file

FUNCTION LoadCourses(fileName)

TRY to open file

IF file can’t open THEN

PRINT "Can't open file"

RETURN

END IF

MAKE empty list called courseList

MAKE empty dictionary called courseDict

WHILE there are more lines in file

READ the line

SPLIT line by commas into parts

IF parts has less than 2 things

PRINT "Bad line, skipping"

CONTINUE

END IF

SET courseNumber = parts[0]

SET courseTitle = parts[1]

MAKE empty list called prerequisites

FOR each item after the title

ADD it to prerequisites

END FOR

MAKE new Course with courseNumber, courseTitle, prerequisites

ADD Course to courseList

ADD Course to courseDict with courseNumber as the key

END WHILE

CLOSE the file

RETURN courseList and courseDict

END FUNCTION

// Check if all prerequisites exist

FUNCTION CheckPrerequisites(courseList, courseDict)

FOR each course in courseList

FOR each prereq in course.prerequisites

IF prereq not in courseDict

PRINT "Missing prerequisite: " + prereq

RETURN false

END IF

END FOR

END FOR

RETURN true

END FUNCTION

// Load courses into a tree

FUNCTION LoadTree(fileName)

CALL LoadCourses(fileName) → courseList, courseDict

IF CheckPrerequisites fails

PRINT "Can't load. Missing prerequisites."

RETURN

END IF

MAKE new binary search tree called tree

FOR each course in courseList

ADD course to the tree

END FOR

RETURN tree

END FUNCTION

// Print one course

FUNCTION ShowCourse(course)

PRINT "Course Number: " + course.courseNumber

PRINT "Course Title: " + course.courseTitle

IF course has prerequisites

PRINT "Prerequisites: " + list of them

ELSE

PRINT "Prerequisites: None"

END IF

END FUNCTION

// Print all courses in order

FUNCTION ShowAllCourses(tree)

IF tree is empty

PRINT "No data loaded."

RETURN

END IF

FUNCTION GoInOrder(node)

IF node is not empty

GoInOrder(node.left)

ShowCourse(node.course)

GoInOrder(node.right)

END IF

END FUNCTION

CALL GoInOrder starting at tree.root

END FUNCTION

*Menu Pseudocode*

START

PRINT "Welcome!"

PRINT "1 - Load course info from file"

PRINT "2 - Show all courses"

PRINT "3 - Show one course and its prerequisites"

PRINT "9 - Quit the program"

WHILE program is running

PRINT "Pick a number from the menu:"

GET userChoice

IF userChoice is 1 THEN

CALL LoadCourses()

ELSE IF userChoice is 2 THEN

CALL ShowAllCourses()

ELSE IF userChoice is 3 THEN

CALL ShowOneCourse()

ELSE IF userChoice is 9 THEN

PRINT "Goodbye!"

STOP program

ELSE

PRINT "That’s not a valid option. Try again."

END IF

END WHILE

END