

**החוג למדעי המחשב
המכללה האקדמית תל-חי**

**מבוא לתכנות מערכות
תרגיל בית מספר 6**

נושאים: עצים, רקורסיה.

סמסטר אביב 2020-21

**תאריך הגשה: 08/06/21, שעה: 23:59
הגשה בזוגות**

בהצלחה!

מטרת התרגיל

עבודה עם עצים. כתיבת פונקציות רקורסיביות.
העבודה הנוכחית תהווה הרחבה של תרגיל בית 5 יחד עם השינויים הדרושים לשימוש בעצים במקום ברשימה מקושרת.

תאור התרגיל

היזמים הצעירים החליטו להוסיף שאילתות רבות למערכת הקיימת ולשם כך התברר לו כי יש לרכוש מערכת משודרגת. אתם מתבקשים לשפר עבורם את המערכת לניהול נתונים אשר נמסרה לפני זמן קצר לידיהם.
אתם מתבקשים לשפר את המערכת באופן שיתואר להלן.

המערכת תכלול ניהול של 3 דברים: ניהול מלאי רכבים, ניהול ספקים, ניהול לקוחות.

1. כל **רכב** מאופיין ע"י 10 שדות:

- מספר רישוי (מספר בן 7 ספרות)
- מספר שלדה (מספר בן 5 ספרות)
- שם היצרן (שם יהיה מורכב ממילה אחת באורך **לא ידוע**)
- שם הדגם (**באורך לא ידוע**)
- צבע (**באורך לא ידוע**)
- שנת יצור (מספר בן 4 ספרות)
- שנת עליה לכביש (מספר בן 4 ספרות)
- מחיר הרכב ששלום לספק (מספר בן עד 7 ספרות)
- המחיר הנוכחי של הרכב (מספר בן עד 7 ספרות)
- נפח מנוע (מספר בן 4 ספרות)

2. כל **ספק** מאופיין ע"י 5 שדות:

- מספר עוסק מורשה (מספר בן 10 ספרות)
- שם הספק (**באורך לא ידוע**)
- טלפון הספק (מספר בן 10 ספרות)
- מספר העסקאות עם הספק (מספר בן 5 ספרות)
- סכום כולל של עסקאות שנעשו עם הספק (מספר בן 10 ספרות)

3. כל **לקוח** מאופיין ע"י 7 שדות:

- שם פרטי (**באורך לא ידוע**)
- שם משפחה (**באורך לא ידוע**)
- ת.ז. (מספר שלם בן 9 ספרות)

- מספר הרישוי של הרכב שהושכר
- תאריך השכרת הרכב (בפורמט: dd/mm/yyyy)
- שעת השכרת הרכב (בפורמט: hh:mm)
- מחיר השכרת הרכב עבור 24 שעות בש"ח (מספר של 3 ספרות)

עליכם לממש את הפעולות הבאות:

1. הגדרת עץ המכוניות **createCarTree** המגדירה עץ ריק של מכוניות.
2. הוספת מכונית חדשה למערכת: **addNewCar** המקבלת מהמשתמש את כל הפרמטרים הדרושים ומגדירה מכונית חדשה במערכת.
3. הגדרת עץ לקוחות החברה **createClientTree** המגדירה עץ ריק של הלקוחות באורך לא ידוע. העץ יוגדר כעץ חיפוש לפי ת.ז. של הלקוח.
4. הוספת לקוח חדש במערכת: **addNewClient** המקבלת מהמשתמש את כל הפרמטרים הדרושים של הלקוח ומגדירה לקוח חדש במערכת.
5. הגדרת עץ ספקי החברה **createSupplierTree** המגדירה עץ ריק של ספקים באורך לא ידוע.
6. הוספת ספק חדש למערכת: **addNewSupplier** המקבלת מהמשתמש את כל הפרמטרים הדרושים של הספק ומגדירה ספק חדש במערכת.
7. שאילתא של מספר הלקוחות של החברה המחזיקים ברכב משנת יצור נתונה **clientNumberWithGivenCarYear** המחזירה את מספר הלקוחות המחזיקים רכבים מאותה שנת יצור נתונה.
8. שאילתא של מספר הרכבים בעלי אותו נפח מנוע: **carNumberWithGivenCapacity** המקבלת מספר המהווה נפח מנוע לבדיקה ומחזירה את מספר הרכבים בחברה בעלי הנפח הנ"ל.
9. מציאת לקוח **findClient** לפי אחד משני פרמטרים: ת.ז. או תאריך השכרת הרכב. אם יש יותר מלקוח אחד, יש למצוא את כולם ולבנות מהם רשימה מקושרת הממוינת לפי מספר ת.ז. יש לבצע פעולה הזאת בצורה רקורסיבית.
(שימו לב כי הכוונה כאן היא לכתוב שתי פונקציות שונות – האחת עבור ת.ז. והשניה עבור התאריך. כל אחת מהן רקורסיבית. ומעליהן להגדיר את הפונקציה **findClient**)
10. שאילתא **averageOfSupplierMoney** המחשבת את הסכום, בממוצע, של כל העסקאות שנעשו עם הספקים.
הערה: יש לבצע פעולה זאת בצורה רקורסיבית. אין להשתמש בפונקציות עזר, משתנים גלובליים או סטטיים.
11. שאילתא של 3 ספקים שאיתם נעשו עסקאות בהיקפים הגדולים ביותר: **threeGreatestSuppliers** המחזירה את מספרי הרישוי של שלושה ספקים איתם נעשו העסקאות כנ"ל.
12. הדפסת רשימת כל הלקוחות עם מספר הרכב שלהם **printClientCarsForGivenRentDate** המקבלת תאריך ומדפיסה את כל הלקוחות ששכרו את רכבם בתאריך זה.
13. הדפסת רשימת כל הספקים עם כל פרטיהם **printSuppliers** המדפיסה את כל הספקים עם כל פרטיהם.

15. מחיקת לקוח מהמערכת **deleteClient** המקבלת מהמשתמש את מספר ת.ז. של הלקוח ומוחקת אותו מהמערכת.
16. מחיקת כל לקוחות החברה מהמערכת **deleteAllClients**.
17. מחיקת רכב מרשימת הרכבים של החברה **deleteCar** המקבלת מהמשתמש את מספר הרישוי ומוחקת אותו מהמערכת.
18. מחיקת רשימת כל הרכבים של החברה **deleteAllCars**.
19. מחיקת ספק מהמערכת **deleteSupplier** המקבלת מהמשתמש את מספר עוסק המורשה של הספק ומוחקת אותו מהמערכת.
20. מחיקת כל ספקי החברה מהמערכת **deleteAllSuppliers**.

דגשים:

יש לבדוק את תקינות הקלט לכל הפונקציות. במקרה שהקלט לא תקין, יש להציג הודעה על שגיאה.

בדקו שהנכם מטפלים גם במקרי קצה.

יש לשמור על קונסיסטנטיות בין המבנים!

יש לתכנן היטב את פתרון התרגיל טרם תחילתו. יש להקפיד על התיכנון הנכון וחלוקת המשימות לקבצים

יש להגיש תוכנית המכילה קבצי מקור (קבצי C) וקבצי header (קבצי h) והן פונקציה ראשית main המדגימה את הבדיקות שנעשו לתוכנה שנכתבה. שימו לב כי פונקציה זאת צריכה להיות קצרה וקריאה!

הודעות שגיאה

סוגי השגיאות עליהן יש לדווח:

קלט לא תקין

הידור, קישור ובדיקה עצמית

- יש לקמפל ולהריץ את התוכנית ב LINUX. שימו לב: תוכנית שלא תתקמפל במערכת הפעלה LINUX תקבל ציון 0!

יש לבצע קומפילציה בשרת המכללה ע"י הרצת הפקודה הבאה:

gcc mySource.c -ansi -Wall -pedantic-errors -lm -o myProg

כאשר mySource.c הוא קובץ או קבצי המקור, ו myProg הוא קובץ ההרצה הנוצר.

דרישות, הגבלות הערות רמזים ותוספות:

- יש להקפיד על תכנון נכון של התוכנית וכתיבה נכונה ב-C.

- יש לתעד את התוכנית.

- בשום אופן אין להשתמש במשתנים גלובאליים!

בפונקציות רקורסיביות:

- אין להשתמש במשתנים סטטיים

- אל תשכחו לבדוק תמיד (!) שהמצביע לעץ שקיבלתם אינו שווה ל NULL.

- למימוש הפונקציות הרקורסיביות עבדו עם שיטת ה"עוזרים". כלומר בקשו מ "עוזר שמאלי" לבצע את הפעולה הרקורסיבית ואחר כך מה "עוזר הימני" לבצע את הפעולה.

- שימו לב כי הרבה פונקציות הקשורות לעצים דומות זו לזו. ולכן מומלץ להתמקד קודם כל בעץ של פריט מסוים. לכתוב ולבדוק היטב את הפונקציות ורק אחר כך לעבור לכתובה של עץ של פריט אחר. הדבר יחסוך לכם הרבה עבודה ותיקונים מיותרים.

הגשת התוכנית:

עליכם להגיש קובץ מכווץ (zipped file) שבו הקבצים המכילים את תוכניתכם (אותם אתם כתבתם) והן קובץ readme עם שמות שני המגשים **למודל**. יש להגיש את העבודה רק מחשבון של אחד השותפים.

שימו לב:

על התרגיל להיות מוגש בזוגות. הנכם רשאים להגיש לבד, אך הדבר אינו מומלץ. עומס התרגיל תוכנן עבור שני סטודנטים. הגשה לבד לא תעניק הקלות.

הגשה באיחור תגרוור קבלת ציון 0 בתרגיל

The errors:

-
- A: Program was not split into separate modules / it was splited badly.
 - B: too long functions (long function bodies). Using long functions makes the code
hard to read, understand and maintain. A well written function shall have a
clearly defined task and shall perform only that task, not several tasks at the same time.
 - C: unnecessary "include"
 - D: didn't check return arguments of the given functions.
 - E: Bad names of variables/functions (not meaningful, limited in length, not in English, etc.)
 - F: No or bad documentation (comments)/ the code is unreadable...
 - G: duplication of code.
 - H: Rare and generally serious errors

ערעורים:

יש להפנות לבודק **אביתר בהלקר** בלבד ע"י שליחת מייל evyatarbhal@gmail.com **תוך שבוע** ממועד פרסום הציונים. בכותרת המייל יש לציין: "ערעור במת"ם עבודת בית מס' X ע"י +ת.ז. של הסטודנטים". יש לקחת בחשבון שבעת הערעור העבודה נבדקת מחדש וכתוצאה מהערעור ציון העבודה עלול להשתנות (יכול לעלות או לרדת). התשובה לערעור תהיה סופית ולא ניתנת לערעור נוסף.