

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

**מבוא לתכנות מערכות
תרגיל בית מספר 4**

נושא: עבודה עם מודולים

סמסטר אביב 2020-21

**תאריך הגשה: 12.05.2021
הגשה בזוגות**

בהצלחה!

מטרת התרגיל

בתרגיל 3 התבקשתם לתכנן משימה לניהול עסק להשכרת מכוניות. בתרגיל זה נממש את המשימה.

שימו לב, בתרגיל זה אין אתם אמורים להשתמש כלל בהקצאה דינאמית. יש להשתמש אך ורק במערכים בגודל קבוע.

תאור התרגיל

יזמים צעירים שמו לב כי מערכת לניהול עסקים של השכרת רכבים טוענת שיפור. הם פנו לאחת מחברות ההשכרה הגדולות בעולם "פרץ" בהצעה להכין עבורה מערכת חדשה לניהול העסק. אולם, כיוון שהיזמים מעולם לא למדו מדעי המחשב, הם החליטו לפנות לחוג למדעי המחשב במכללה ולבקש מהסטודנטים המוכשרים בחוג לבצע עבורם את המשימה תמורת תשלום הולם. הסטודנטים נענו בשמחה וקיבלו את הדרישות הבאות מהיזמים.

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 (באורך עד 10 תווים)
- שם הדגם (באורך עד 10 תווים)
- צבע (באורך עד 7 תווים)
- שנת יצור (מספר בן 4 ספרות)
- שנת עליה לכביש (מספר בן 4 ספרות)
- מחיר הרכב ששולם 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 (בן 5 תווים)
- 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 עד 20 תווים)
 - שם משפחה (באורך עד 20 תווים)
 - 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 בן 3 ספרות)

עליכם לממש את הפעולות הבאות:

1. הגדרת רשימת המכוניות **createCarList** המגדירה מערך של מכוניות בגודל N (כאן N הוא מספר המכוניות המקסימלי שיכולות להשתייך לחברת ההשכרה. לשם פשטות ניתן להניח כי $N = 10000$) ומאפסת את המבנה.
2. הוספת מכונית חדשה למערכת: **addNewCar** המקבלת מהמשתמש את כל הפרמטרים הדרושים ומגדירה מכונית חדשה במערכת.
3. הגדרת רשימת לקוחות החברה **createClientList** המקבלת מהמשתמש מספר M ומגדירה מערך של לקוחות החברה בגודל M (כאן M הוא מספר הלקוחות המקסימאלי שהחברה מסוגלת לקלוט) ומאפסת את המבנה. לשם פשטות ניתן להניח כי $M = 8000$.
4. הוספת לקוח חדש במערכת: **addNewClient** המקבלת מהמשתמש את כל הפרמטרים הדרושים של הלקוח ומגדירה לקוח חדש במערכת.
5. הגדרת רשימת ספקי החברה **createSupplierList** המקבלת מהמשתמש מספר K ומגדירה מערך של ספקי החברה בגודל K (כאן K הוא מספר מקסימאלי של ספקי החברה) ומאפסת את המבנה. לשם פשטות ניתן להניח כי $K = 50$.
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. שאילתא של 3 ספקים שאיתם נעשו עסקאות בהיקפים הגדולים ביותר: **threeGreatestSuppliers** המחזירה את מספרי הרישוי של שלושה ספקים איתם נעשו העסקאות כנ"ל.
10. מחיקת לקוח מהמערכת **deleteClient** המקבלת מהמשתמש את מספר ת.ז. של הלקוח ומוחקת אותו מהמערכת.

11. מחיקת כל לקוחות החברה מהמערכת **deleteAllClients**.
12. מחיקת רכב מרשימת הרכבים של החברה **deleteCar** המקבלת מהמשתמש את מספר הרישוי ומוחקת אותו מהמערכת.

13. מחיקת רשימת כל הרכבים של החברה **deleteAllCars**.
14. מחיקת ספק מהמערכת **deleteSupplier** המקבלת מהמשתמש את שם הספק ומוחקת אותו מהמערכת.

15. 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. בשלב זה של לימודיכם הנכם מסוגלים ונדרשים להיות מסוגלים לתכנן ולכתוב תוכנית בסדר הגודל הנתון בעצמכם. תכנון התוכנית מהווה חלק מהותי מהתרגיל. הקפידו לבצע חלוקה של התוכנית למודולים ופונקציות בהתאם למבנה הלוגי של התוכנית ולא לכתוב את התוכנית כפונקציה main אחת גדולה. לדוגמא, נצפה לראות הפרדה בין הפונקציות המטפלות בקלט, לבין אילו המנהלות את מבני הנתונים השונים.
- יש לתעד את התוכנית. את אופן התייעוד אנו מניחים לכם לקבוע בעצמכם. ההנחיה היחידה שתיתן הנה כי מי שרוצה להיעזר בפונקציות שאתם כותבים, צריך להיות מסוגל לבצע זאת ע"י הסתכלות בהצהרת הפונקציה וקריאת התייעוד ללא צורך בקריאת גוף הפונקציה.
- הקפידו להגדיר פונקציות המחזירות ערך. כלומר הימנעו מפונקציות המחזירות void. הדבר נכון גם לפונקציה main.

- 4 תוודאו שבכל יציאה מהפונקציה, היא מחזירה ערך.
- 5 שימו לב כי בתרגיל זה אנחנו לא מתשמשים בהקצאה דינאמית. אין שום צורך בה, כי כל המערכים נתונים בגודל קבוע מראש ויש להגדיר אותם כמשתנים גלובליים. זה אומר שפונקציות ה- Create (כמו CreateCarList) לא באמת מגדירות את המבנה, אלא רק מאפסות אותו.
- 6 המלצה חמה לגבי פונקציות ה- Add. נקח לדוגמה את **addNewCar** : כדי להקל על העבודות העתידיות, מומלץ לבנות 3 פונקציות עזר:
 - פונקציה המבקשת מהמשתמש את פרטי המכונית ומחזירה מכונית בודדת.
 - פונקציה הבודקת את תקינות הקלט, כלומר שאכן כל השדות של המכונית הוגדרו באופן תקין.
 - פונקציה המקבלת מכונית ומכניסה אותה למבנה. במקרה שלנו לתוך מערך המכוניות. לאחר מכן addNewCar רק קוראת ל 3 פונקציות העזר ולא מבצעת דבר נוסף.
- 7 מומלץ לא לכתוב את כלל הפונקציות ורק אז לגשת לבדיקות והרצות, אלא לכתוב פונקציה אחת, לדבג, לבדוק אותה היטב ורק אז לעבור לפונקציה הבאה. באופן כזה, תוכלו לחסוך הרבה מאוד עבודה, מכיוון שחלק מהפונקציות דומות ולכן אם תסיימו היטב פונקציה אחת, שאר הפונקציות יצטרכו שינויים קטנים בלבד.

הגשת התוכנית:

עליכם להגיש קובץ מכוץ (zipped file) שבו הקבצים המכילים את תוכניתכם (אותם אתם כתבתם) והן קובץ readme עם שמות שני המגישים **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 תגרור 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.