

**תרגיל 5 – רקורסיה, מבנים וקבצים****הגשה עד 13.01.2019 בשעה 23:50****הוראות הגשה:**

1. שאלות ובקשות בקשר לעבודה להפנות אך ורק למרצה גב' סבטלנה רוסין, במייל: [sceassign2016@gmail.com](mailto:sceassign2016@gmail.com).
2. תרגילים הם ביחידים! כל עבודה משותפת היא אסורה ותיענש בחומרה!
3. להגיש רק תכניות שעוברות קומפילציה על מהדר שפת C gcc Ubuntu Linux.
4. ההגשה היא של קבצי הקוד (קובץ c). בלבד. יש ליצור 3 קבצים: part1.c, part2.c ו-part3.c, לכוון את כל הקבצים לקובץ אחד בפורמט RAR או ZIP, ולהגיש רק קובץ זה.
5. בתחילת הקובץ יש להוסיף את התיעוד הבא:
 

```
/* Assignment: 5
   Campus: Ashdod / Beer Sheva (תבחרו את המתאים)
   Author: Israel Israeli, ID: 01234567
*/
```

כמובן שיש לעדכן את השמות ומספרי תעודות הזהות שלכם.
6. הארכות יינתנו אך ורק במקרים חריגים (מילואים, אבל על קרובים ומחלה חריפה!) ובצרוף אישורים מתאימים. כמו כן במקרה של ידע מוקדם חובה ליצור קשר עם המרצה האחראית על התרגיל לפחות יומיים לפני חלוף הדד-ליין!
7. ההגשה היא עד התאריך האחרון לתרגיל: 13/01/19 בשעה 23:50. הגשה מאוחרת אפילו בדקה – לא תתקבל (המערכת חוסמת את אפשרויות ההגשה!). קחו זאת בחשבון ותכננו את זמנכם בהתאם!
8. אין להשתמש בתרגיל בחומר שטרם נלמד, או שנלמד לאחר נושא התרגיל, אלא אם נכתב במפורש בתרגיל שמותר.
9. הקלטים יהיו מהטיפוסים החוקיים. ז"א בכל מקום שצריך להכניס מספר שלם – נכניס מספר שלם (ולא שבר או אות). אנחנו לא מתחייבים שהוא יהיה חיובי או א-שלילי, או בטווח מסוים – אלא אם כן נאמר אחרת בשאלה עצמה.
10. בכל פעם שהמשתמש מקליד קלט שגוי התוכנית מבקשת קלט חוזר.
11. אחרי כל הדפסה יש לבצע ירידת שורה.
12. בתרגיל יש להשתמש בספריית malloc, stdlib, string, stdio בלבד!
13. יש להקפיד על תכנות נכון:
  - a. כל הערכים שהם קבועים, (מבחינה לוגית הם לא אמורים להשתנות), חייבים להיות מוגדרים כ: const, define או enum, בהתאם לצורך.

- b. יש לרשום הערות .
- c. יש להקפיד על הזחות!!! כיתוב נכון וקריא! ושמות משמעותיים!
- d. יש לנסות ולייעל את הקוד והתוכנית ככל שניתן.
- e. לפני בקשת קלט (scanf) יש להדפיס למשתמש הוראה (printf) איזה קלט מבוקש.
- f. יש להקפיד על מוסכמות התכנות הנכון (שמות כמו שצריך וכו').
- g. יש להקפיד על כל כללי התכנות הנכון כפי שנלמדו בכיתה.

בהצלחה 😊

**חלק א': פונקציות רקורסיביות (45 נקודות) אסור להשתמש במשתנים סטטיים (static) וגלובליים !**

כתוב את הפונקציות הבאות באופן **רקורסיבי**. הפונקציות לא משנות את המערכים שהן מקבלות כפרמטרים, מלבד פונקציה מס' 5.

1. ( 5 נק' ) פונקציה המקבלת כפרמטר מספר שלם חיובי או שלילי, ומחזירה את הספרה הקטנה ביותר. למשל,  
הפונקציה מקבלת מספר 35672 ומחזירה 2.  
הפונקציה מקבלת מספר -31672 ומחזירה 1.
2. ( 5 נק' ) פונקציה המקבלת כפרמטר מספר שלם בבסיס דצימלי (10) ומחזירה את המספר בבסיס בינארי (2) . למשל, הפונקציה מקבלת מספר 6 ומחזירה 110.
3. ( 5 נק' ) פונקציה המקבלת כפרמטרים מערך של שלמים ואת גודלו. הפונקציה מחזירה אמת אם כל אברי המערך אי זוגיים. אחרת, היא מחזירה שקר. למשל,  
הפונקציה מקבלת מערך { 1, 3, 7, 9, 2, 5 } ומחזירה שקר  
הפונקציה מקבלת מערך { 1, 3, 7, 9, 5, 5 } ומחזירה אמת
4. ( 5 נק' ) פונקציה מקבלת מערך וגודלו ומחזירה את סכום האיברים הזוגיים הכלולים בו,  
לדוגמה , מקבלת מערך :  
7,9,6,9,2,4  
ומחזירה : 12
5. ( 5 נק' ) פונקציה המקבלת כפרמטרים מערך של שלמים ואת גודלו, ומשנה את המערך כך שהוא יכיל את סידרת הסכומים ההתחלתיים של המערך המקורי. לדוגמא, אם בתחילת הפונקציה המערך הוא {1,2,3,4,5} אז בסיום הפונקציה המערך הוא {1,3,6,10,15}.

6. ( 2 נק' ) פונקציה מקבלת מחרוזת ומחזירה את כמות התווים (אורך) המופיעים בה, לא כולל '\0'.

7. ( 5 נק' ) פונקציה מקבלת שתי מחרוזות, משווה אותם לקסיקוגרפית (בדומה לפונקציה מוכנה strcmp) ומחזירה 0 במידה והמחרוזות זהות, ערך חיובי כאשר מחרוזת הראשונה גדולה מהשנייה, אחרת מחזירה ערך שלילי (אסור להשתמש בפונקציות מוכנה!).

8. ( 8 נק' ) פונקציה המקבלת כפרמטר מערך מחרוזות ואת גודלו, ומחזירה את האורך המקסימלי של מחרוזת במחרוזת הארוכה ביותר, ואת מספר המחרוזות במערך בעלות אורך מקסימלי.

למשל, עבור מערך מחרוזות באה :

```
char* text[] = {"test", "correct", "data", "one", "replace"};
```

הפונקציה תחזיר : 7 ו-2

הערות לפונקציה מס' 8 : מותר להוסיף פרמטרים לפונקציה.

מותר להיעזר בפונקציית עזר – אך עליה להיות רקורסיבית גם כן.

### הפונקציה הראשית (5 נקודות)

כתוב את הפונקציה הראשית בתכנית. הפונקציה הראשית קוראת לפונקציות הרקורסיביות מחלק א' בזו אחר זו עם קלט שהמשתמש מקליד, ומדפיסה את הערכים המוחזרים. ניתן להניח כי האורך של כל מחרוזת אינו עולה על 80 תווים.

הערה : בהפעלת הפונקציה בוליאנית (מס' 3) התוכנית תדפיס הודעה מתאימה לפי הבדיקה, למשל במידה והפונקציה מחזירה אמת התוכנית תדפיס "All digits of the number are odd", אחרת תדפיס "Not all digits of the number are odd".  
עבור פונקציה מס' 8 מותר ליצור מערך המחרוזות ע"י שורת האתחול כמו בדוגמה למעלה.

### חלק ב': מבנים – מענה בקובץ part2.c 40 נק':

א. ( 2 נק' ) הגדר מבנה Employee, המתאר נתונים של עובד במפעל שבו 5 מחלקות. (מספרי המחלקות הם 1-5). המבנה מכיל את השדות הבאים: שם העובד (מחרוזת דינאמית – המיוצגת על-ידי מצביע), מספר מחלקה, משכורת מסוג מספר ממשי.

ב. ( 10 נק' ) כתוב פונקציה

```
Bool initEmployee(struct Employee *employee)
```

הקולטת מהמסך נתונים עבור עובד אחד במפעל. יש לבצע בדיקת קלט עבור מספר המחלקה, במידה והקלט לא היה תקין יש לדרוש הזנה חוזרת. במידה והזיכרון דינאמי (עבור משתנה שם) לא יתקבל, הפונקציה תחזיר שקר, אחרת אמת.

ג. ( 10 נק' ) כתוב פונקציה

```
void calcAverages(struct Employee *data, int size, float *result)
```

הפונקציה מקבלת כפרמטרים מערך של מבנים מסוג Employee, מספר העובדים במפעל, וכן מערך נוסף בגודל 5. עבור כל  $0 \leq i \leq 4$  הפונקציה כותבת ב-`result[i]` את השכר הממוצע של מחלקה  $i + 1$ . במידה ומחלקה אינה מכילה אף עובד יש להכניס למערך ערך -1.  
ד. (10 נק') כתוב פונקציה

int buildFactory(struct Employee \*\* pdata)

הפונקציה מקבלת מצביע למערך של מבנים מסוג Employee וקולטת את כמות העובדים במפעל, יוצרת מערך דינאמי, מאתחלת אותו ומחזירה את כמות העובדים במפעל.

ד. (8 נק') כתוב את הפונקציה הראשית בתכנית. הפונקציה מבצעת את הפעולות הבאות:

- יוצרת מערך דינאמי של העובדים במפעל.
- מדפיסה את פרטי העובדים.
- מחשבת את ממוצע המשכורות בכל מחלקה.
- מדפיסה את ממוצע המשכורות של כל מחלקה. עבור מחלקה שאין בה עובדים יש להדפיס הודעה מתאימה.
- לבסוף התוכנית משחררת את כל הזיכרון הדינאמי.

הערה:

ניתן ומומלץ להוסיף פונקציות עזר.

ניתן להניח כי האורך של כל שם אינו עולה על 20 תווים.

### חלק ג': קבצי טקסט – מענה בקובץ part3.c 15 נק':

א. (2 נק') הגדר מבנה `TaskTime` המתאר את משך הזמן הדרוש לביצוע משימה, ומכיל שעות, דקות ושניות.

ב. (3 נק') הגדר מבנה `Marathon` המתאר את הזמן שבו ספורטאי רץ מרתון, ומכיל את השדות הבאים: שם של ספורטאי (באורך 10 תווים לכל היותר), מספר תעודת זהות, זמן ריצה של מרתון מסוג `TaskTime`.

ד. (10 נק') כתוב את הפונקציה הראשית בתכנית. הפונקציה מבצעת את הדברים הבאים:

- קולטת מהמשתמש שם של קובץ קלט ושם של קובץ פלט.
- קוראת נתונים מהקובץ וכותבת לקובץ הפלט את כל הנתונים על הספורטאי המהיר ביותר. במידה וקיימים מספר ספורטאים בעלי תוצאת שיא דומה (הכי מהירים), הפונקציה כותבת את כולם.

דוגמא לקובץ קלט:

Eli 025556660 3 44 55  
Beni 310717482 4 2 55  
Yonatan 029626543 3 44 2  
Yossi 012345678 4 55 2  
Ruti 987654321 3 44 2

הערות :

ניתן להניח כי שם של ספורטאי לא מכיל רווחים.

ניתן ומומלץ להוסיף פונקציות עזר