

## תרגיל 5

הגשה עד 26.1.2017 בשעה 23:50

הוראות ההגשה ודגשים מיוחדים מופיעים בסוף התרגיל! **חובה לקרוא ולפעול לפיהן.**

### חלק א': פונקציות רקורסיביות (70 נקודות)

כתוב את הפונקציות הבאות **באופן רקורסיבי**. הפונקציות לא משנות את המערכים שהן מקבלות כפרמטרים. מותר להוסיף פרמטרים לפונקציות שלא מפורטת בהן חתימה מדוייקת.

1. פונקציה

`long repeat(int digit, int length)`

ניתן להניח ללא בדיקה כי הפרמטר digit הוא שלם בין 0 ל-9. הפונקציה מחזירה מספר שלם עם length ספרות שכל אחת מהן היא digit. לדוגמא, אם  $\text{digit}=3$  ו- $\text{length}=5$  אז הערך המוחזר הוא 33333. אם  $\text{digit}=0$  ו- $\text{length}=5$  אז הערך המוחזר הוא 0.

2. פונקציה המקבלת כפרמטר מערך של מספרים שלמים וגודלו. הפונקציה מחזירה 1 אם כל המספרים במערך חיוביים. אחרת, היא מחזירה 0. לדוגמא, אם המערך הוא {2,7,13,51,77} אז הפונקציה מחזירה 1. עבור המערך {2,3,0,0,48,49} יוחזר ערך 0.

3. פונקציה המקבלת כפרמטר מערך של מספרים שלמים וגודלו. הפונקציה מחזירה 1 אם לפחות אחד מהמספרים במערך שלילי. אחרת, היא מחזירה 0. לדוגמא, אם המערך הוא {2,-7,-13,51,77} אז הפונקציה מחזירה 1. עבור המערך {2,3,0,0,48,49} יוחזר ערך 0.

4. פונקציה המקבלת כפרמטר מספר שלם. הפונקציה מחזירה 1 אם הספרות של המספר ממוינות בסדר עולה ממש. אחרת, היא מחזירה 0. לדוגמא, עבור כל אחד מהמספרים 0, 5, 3578, 14689 יוחזר 1. עבור כל אחד מהמספרים 506, 3779, 741 יוחזר 0.

5. פונקציה המקבלת כפרמטרים מערך של של ציונים וגודלו. הפונקציה מחזירה את כמות הציונים העוברים. (לפחות 56). לדוגמא, אם המערך הוא {100,81,49,32,56,81,56,49} אז הפונקציה מחזירה 5.

6. פונקציה המקבלת כפרמטר מחרוזת ותו, ומחזירה את מספר הפעמים שהתו מופיע במחרוזת. לדוגמא, אם המחרוזת היא "one two three four": והתו הוא 'e' אז הערך המוחזר הוא 3.

7. פונקציה המקבלת כפרמטר מחרוזת, ומדפיסה אותה תו אחר תו לפי הסדר המקורי.

## חלק ב': הפונקציה הראשית (5 נקודות)

כתוב את הפונקציה הראשית בתכנית. הפונקציה 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. עבור כל מערך שאינו מחרוזת מוקצה זיכרון באופן דינמי בגודל מדויק שהמשתמש מבקש. יש לשחרר את כל הזיכרון הדינאמי בסיום הפונקציה הראשית.

## חלק ג': מבנים (25 נקודות)

נתונה ההגדרה:

```
struct date
{
    int day, month, year;
};
```

המתארת תאריך. כמוכן נתונה ההגדרה הבאה המתארת סטודנט:

```
struct student
{
    char name[15];
    struct date birth;
    int courses;
    int *grades;
};
```

שדה courses מכיל את כמות הקורסים שהסטודנט למד וקיבל בהם ציון. שדה grades הוא מערך של ציונים באורך courses.

סטודנט מצטיין הוא סטודנט שלמד וקיבל ציון בלפחות 3 קורסים, כל ציוניו הם לפחות 80, והממוצע שלו לפחות 90.

כתוב תכנית הכוללת את הפונקציות הבאות:

- פונקציה `int initialize(struct student **pdata)` המתחלת באמצעות קלט מהמשתמש מערך של מבנים, ומחזירה את גודלו.
- פונקציה המקבלת כפרמטרים מערך של סטודנטים ותאריך נוכחי. הפונקציה מדפיסה על המסך את השמות של כל הסטודנטים שיש להם היום יום הולדת, ואת גיליהם.
- פונקציה המקבלת כפרמטר שם של קובץ טקסט, פותחת אותו לכתיבה, מדפיסה לתוכו את השמות של כל הסטודנטים המצטיינים, וסוגרת את הקובץ. הפונקציה מחזירה את כמות הסטודנטים שפרטיהם הודפסו בקובץ. אם פתיחת הקובץ נכשלה יוחזר 1-.
- פונקציה הראשית בתכנית הקולטת מהמשתמש את התאריך הנוכחי ושם של קובץ פלט, וקוראת לשלושת הפונקציות האחרות. הפונקציה מדפיסה את מספר הסטודנטים המצטיינים, או הודעה מתאימה במקרה של כישלון בפתיחת הקובץ. כמו כן, הפונקציה משחררת את הזיכרון הדינמי שהוקצה בתכנית.

מותר להוסיף פונקציות לפי הצורך.

## חלק ד': סעיף בונוס (מסלול סייבר) (20 נקודות)

נתונה ההגדרה הבאה של איבר ברשימה מקושרת:

```
typedef struct item
{
    int num;
    struct item *next;
} item;
```

- א. כתוב פונקציה שמקבלת מצביע לראש רשימה מקושרת כזו, ומחזירה את המספר המינימלי שמופיע ברשימה. אם הרשימה ריקה אז יוחזר 0.
- ב. כתוב את הפונקציה הראשית בתכנית. הפונקציה הראשית מבצעת את הפעולות הבאות:
- קולטת מהמשתמש 5 מספרים.
  - בונה באמצעות הקלט של המשתמש רשימה מקושרת של 5 איברים, שמוקצים באופן דינמי. האיברים מופיעים ברשימה בסדר שהקליד אותם המשתמש. מספר יכול להופיע ברשימה יותר מפעם אחת.
  - קוראת לפונקציה מסעיף א', ומדפיסה את הערך המוחזר.
  - משחררת בצורה מסודרת את הזיכרון שהוקצה באופן דינמי.

## הערות:

### 1. אין להשתמש בעבודה זו במשתנים סטטיים וגלובליים.

2. 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.
3. בכל פעם שהמשתמש מקליד קלט שגוי התוכנית מבקשת קלט חוזר.
4. אחרי כל הדפסה יש לבצע ירידת שורה.
5. יש להקפיד על תכנות נכון:
- a. כל הערכים שהם קבועים, (מבחינה לוגית הם לא אמורים להשתנות), חייבים להיות מוגדרים כ: `enum` או `const`, `define`, 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 כל כללי התכנות הנכון כפי שנלמדו בכיתה.
6. בהצלחה ☺

## הנחיות הגשה

- תרגילים הם ביחידים! כל עבודה משותפת היא אסורה ותיענש בחומרה!
- כל הפונקציות של סעיפים א', ב', נכתבים כחלק של אותה תוכנית וזה מצריך פונקציה ראשית אחת בלבד. עבור חלק ג' יש לכתוב תוכנית נוספת, ועוד תוכנית עבור חלק ד'. ליצור קובץ ZIP שמכיל את כל קבצי C.
- 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.

- שאלות ובקשות בקשר לעבודה להפנות אך ורק למרצה האחראי, **דוד טנקוס**, במייל: [davidt@sce.ac.il](mailto:davidt@sce.ac.il)
- 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!
- ההגשה היא עד התאריך האחרון לתרגיל: **26.1.2017** בשעה **23:50**. הגשה מאוחרת אפילו בדקה – לא תתקבל (המערכת חוסמת את אפשרויות ההגשה!). קחו זאת בחשבון ותכננו את זמנכם בהתאם!