תרגיל 5 – רקורסיה, מבנים וקבצים

הגשה עד 27.01.2020 בשעה 23:50

הוראות הגשה:

1. שאלות ובקשות בקשר לעבודה להפנות אך ורק למרצה גב' סבטלנה רוסין, במייל: 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 (כקובץ .c) בלבד. יש ליצור 3 קבצים : part1.c, part2.c ו- part3.c, לכווץ את כל הקבצים לקובץ אחד בפורמט RAR או ZIP, ולהגיש רק קובץ זה.
5. בתחילת הקובץ יש להוסיף את התיעוד הבא:

```
/* Assignment: 5
        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: 27/01/20 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 בספריות stdio, string, stdlib, malloc בלבד!
13. יש להקפיד על תכנות נכון:
    a. כל הערכים שהם קבועים, (מבחינה לוגית הם לא אמורים להשתנות), חייבים להיות מוגדרים כ: define, const 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 !

כתבו את הפונקציות הבאות באופן **רקורסיבי**. הפונקציות לא משנות את המערכים שהן מקבלות כפרמטרים , מלבד פונקציה מס' 6.

1. (5 נק') פונקציה ***SumRange*** *המקבלת* כפרמטר שני מספרים שלמים low ו-high *ומחזירה* את סכום כל השלמים בקטע [low,..,high]. למשל, אם high=15, low=10, אז הערך המוחזר הוא:

75 = 10+11+12+13+14+15

ניתן להניח ללא בדיקה כי low ≤ high.

2. (5 נק') פונקציה בולאנית בשם ***OnlyEven*** המקבלת מספר שלם *ומחזירה* אמת (1) במידה וכל הספרות של המספר הינן זוגיות , אחרת מחזירה שקר (0). למשל,
הפונקציה מקבלת מספר 6812 ומחזירה שקר
הפונקציה מקבלת מספר 2684 ומחזירה אמת

3. (5 נק') פונקציה בשם ***RemoveDigit*** *המקבלת* מספר שלם ארוך וסיפרה, הפונקציה *מחזירה* את המספר המתקבל אחרי ההורדה של כל המופעים של הספרה במספר. במידה והסיפרה לא מופיעה כלל, הפונקציה תחזיר את המספר המקורי.
למשל :
הפונקציה מקבלת מספר 3527267 וספרה 7 ומחזירה מספר 35226
הפונקציה מקבלת מספר 2222 וספרה 2 ומחזירה מספר 0
הפונקציה מקבלת מספר 1452 וספרה 3 ומחזירה מספר 1452

4. (5 נק') פונקציה בולאנית ***CheckArrays*** המקבלת כפרמטרים שני מערכים ואת גודלם. המערכים באותו הגודל – הפונקציה *מחזירה* אמת אם כל מערך שווה למערך השני בסדר הפוך. אחרת, היא מחזירה שקר.
למשל ,הפונקציה מקבלת שני מערכים בגודל 7 כל אחד:

first[] = { 7, 36, 125, 321, 245, 20, 1 }
second[] = { 1, 20, 245, 321, 125, 36, 7 }

הפונקציה מחזירה אמת

המכללה האקדמית להנדסה סמי שמעון
קמפוס באר שבע ביאליק פינת בזל 84100 | קמפוס אשדוד ז'בוטינסקי 84, 77245 | www.sce.ac.il | חייג: *מהפלפון

5. (5 נק') פונקציה בולאנית ***IsInArray*** מקבלת מערך של מספרים שלמים כולל את גודלו וערך נוסף. הפונקציה בודקת האם הערך מופיע במערך, במידה ואכן מופיע *הפונקציה מחזירה* אמת (1), אחרת מחזירה שקר (0). למשל,
הפונקציה מקבלת מערך {4,5,7,2} , ערך נוסף 2 ומחזירה אמת
הפונקציה מקבלת מערך {4,5,7,2}, ערך נוסף 10 ומחזירה שקר

6. (5 נק') פונקציה ***UpdateArray*** מקבלת מערך של מספרים שלמים כולל את גודלו וערך נוסף. הפונקציה כופלת כל איבר במערך פי מספר הזה. למשל,
הפונקציה מקבלת מערך {4,5,7,2} וערך נוסף 10

הפונקציה מעדכנת מערך ל:

{40,50,70,20}

7. (5 נק') פונקציה ***FirstOccurenceUpper*** מקבלת מחרוזת *ומחזירה* את המצביע להופעה הראשונה של תו שהוא אות לטינית גדולה. במידה ואות גדולה לא מופיעה כלל, הפונקציה מחזירה NULL . למשל,

```
char str[] = "testThisString"
```

הפונקציה תחזיר מצביע לאות 'T' (תכנית הראשית תדפיס את התו).

8. (5 נק') פונקציה בולאנית ***CheckString*** המקבלת כפרמטר מצביע לתחילת המחרוזת ומצביע לאמצע המחרוזת ובודקת האם חצי הראשון שלה זהה לשני ללא התייחסות לאות קטנה וגדולה, כלומר אות 'a' ו- 'A' נחשבות זהות. הפונקציה *מחזירה* אמת (1) במידה וקיים שיוון אחרת היא *מחזירה* שקר (0). למשל,

הפונקציה מקבלת מחרוזת " PeacEpeace" ומחזירה אמת

הפונקציה מקבלת מחרוזת "PeacePwar" ומחזירה שקר

<u>הפונקציה הראשית (5 נקודות)</u>

כתבו את הפונקציה הראשית בתכנית. הפונקציה הראשית קוראת לפונקציות הרקורסיביות מחלק א' בזו אחר זו . ניתן ליצור ולאתחל את כל המערכים ומחרוזות בשורת האתחול (ללא קלט מהמשתמש). התוכנית מדפיסה את הערכים המוחזרים. ניתן להניח כי האורך של כל מחרוזת אינו עולה על 80 תווים.

<u>הערה</u> : בהפעלת הפונקציות בוליאניות (מס' 2,4,5,8) התוכנית תדפיס הודעה מתאימה לפי הבדיקה, למשל במידה והפונקציה מסי 5 מחזירה אמת התוכנית תדפיס
"The value appears in the given array"
אחרת תדפיס:
."The value does not appear in the given array"

המכללה האקדמית להנדסה סמי שמעון
קמפוס באר שבע ביאליק פינת בזל 84100 | קמפוס אשדוד ז'בוטינסקי 84, 77245 | www.sce.ac.il | חייג: *מטמבמט

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

א. נתונות ההגדרות הבאות :
מבנה *Course* המכיל מידע על קורס שלמד סטודנט : שם הקורס והציון:

```
struct Course {
    char* courseName ;
    int grade;
};
```

מבנה *Student* המכיל מידע על סטודנט : שם הסטודנט , מערך של קורסים שלמד ואת גודלו של המערך:

```
struct Student {
    char* studentName ;
    struct Course * course_arr ;
    int size;
};
```

ניתן להניח כי האורך של שם של סטודנט או של קורס אינו עולה על 30 תווים. שמות אלו יכולים להכיל רווחים (יותר ממילה אחת בכל שם). יש להקצות לכל שם זיכרון באופן מדוייק.

ב. כתבו את הפונקציות הבאות:

void **InputCourse**(Course* pCourse)
(6 נק') הקולטת מהמסך נתונים עבור קורס אחד לסטודנט.

void **PrintCourse**(Course Cours)
(2 נק') המדפיסה על מהמסך נתונים עבור קורס אחד לסטודנט.

void **InputStudent**(Student* pSt)
(8 נק') הקולטת מהמסך נתונים עבור סטודנט אחד, כולל נתונים על הקורסים של הסטודנט.

void **PrintStudent**(Student st)
(3 נק') המדפיסה על מהמסך נתונים עבור סטודנט אחד, כולל נתונים על הקורסים של הסטודנט.

int **Build**(Student** pSt)
(7 נק') יוצרת ומאתחלת מערך של מבנים מסוג Student באמצעות קלט מהמשתמש. היא מחזירה את המערך (דרך המצביע) ואת גודלו. ניתן להניח ללא בדיקה כי הקלט תקין : כל שם של סטודנט מופיע לכל היותר פעם אחת במערך.

void **FreeAll**(Student** pSt, int size)
(6 נק') משחררת את כל הזיכרון הדינאמי שהוקצא.

void **PrintAll**(Student* arrSt, int size)

(3 נק') המדפיסה את כל הפרטים של הסטדונטים, כולל פרטים על הקורסים שלקחו.

void **CourseList**(Student* arrSt, int size, char* CourseName);
(5 נק') המקבלת כפרמטרים מערך של סטודנטים ואת גודלו וכמו-כן את שם הקורס. הפונקציה מדפיסה את השמות של כל הסטודנטים שעברו קורס זה (הציון שלהם בקורס הוא לפחות 56).

ניתן להשתמש ברייבר (main) הקיים :

```
int main(){
      Student* arr=NULL;
      int size;
      size = Build(&arr);
      PrintAll(arr, size);
      CourseList(arr, size, "os");
      FreeAll(&arr, size);
      return 0;
}
```

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

כתבו פונקציה הקוראת נתונים מקובץ טקסט בשם passwords.txt המכיל סיסמאות, כאשר כל סיסמא בקובץ מופיעה בשורה נפרדת וכותבת לקובץ פלט בשם weakpass.txt את הסיסמאות ה"חלשות". הסיסמא ה-"חלשה" היא בעלת ציון הנמוך מ- 4.

הסיסמא מכילה אותיות (קטנות או/ו גדולות), ספרות והסימנים המיוחדים הבאים: @ ו- % בלבד.

כללי חישוב הציון הם:

1) אם הסיסמא מכילה גם אותיות גדולות וגם אותיות קטנות – תוספת 2 נקודות לציון.
2) אם קיימת לפחות סיפרה אחת – תוספת של 2 נקודות לציון לסעיף 1.
3) אם קיים לפחות סימן מיוחד אחד (@ או- %) – תוספת של 3 נקודות לציון סעיפי 1 ו-2.
4) במידה ואורך הסיסמא ארוך יותר מ-6 תווים – תוספת של 3 נקודות לציון לכל הסעיפים הקודמים. למשל, הקובץ passwords.txt:

aBa
x@Y2ykk

ציון הסיסמא הראשונה הינו 2 (סעיף 1)

ציון הסיסמא השנייה הינו 10 = 3+3+2+2

לקובץ weakpass.txt ירשם:

aBa

יש לכתוב את התוכנית הראשית שמפעילה את הפונקציה.
הערה: יש להכין קובץ password.txt מראש ולשמור אותו ביחד עם קובץ part3.c