

תרגיל 5 – רקורסיה מבנים וקבצים

הגשה עד 2/2/2025 בשעה 23:50

הוראות הגשה:

1. עבור כל חלק ליצור קובץ C עם סיומת c. חובה לתת שמות לקבצים כ-
part1.c, part2.c, part3.c.
2. לכוון את כל הקבצים לקובץ אחד בפורמט RAR או ZIP, ולהגיש רק קובץ זה.
3. להגיש רק תכניות שעוברות קומפילציה על **מחדר שפת Visual Studio C**.
4. ההגשה רק דרך מערכת מודול באופן עצמאי, ולא בזוגות או בקבוצות גדולות יותר.
5. יש לכתוב הערות באנגלית בלבד. יש להקפיד על ההזחות, כיתוב נכון וקריא, שמות משמעותיות.
6. שאלות ובקשות בקשר לעבודה **להפנות אך ורק לגב' סבטלנה רוסין**, במייל:
sceassign2016@gmail.com
7. בתחילת כל קובץ יש להוסיף את התיעוד הבא:
/* Assignment: 5
Author: Israel Israeli, ID: 01234567
*/
כמובן שיש לעדכן את השמות ומספרי תעודות הזהות שלכם.
8. 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 האחראי על התרגיל לפחות יומיים לפני חלוף הדד-ליין!
9. ההגשה היא עד 2/2/2025 בשעה 23:50. הגשה מאוחרת אפילו בדקה – לא תתקבל. (המערכת חוסמת את אפשרויות ההגשה!) קחו זאת בחשבון ותכננו את זמנכם בהתאם!

חלק א' – רקורסיה – מענה בקובץ part1.c – 40 נק'

כתבו את הפונקציות הבאות **באופן רקורסיבי**. הפונקציות לא משנות את המערכים שהן מקבלות כפרמטרים. הפתרון לכל סעיף הוא פונקציה אחת קצרה ללא פונקציות עזר. חובה לשמור על שמות הפונקציה המוגדרים. **אסור להשתמש במשתנים static**.

1. (5 נק') פונקציה
`void printNumers(int begin, int end, int skip)`
הפונקציה מדפיסה את כל המספרים מ-begin עד end בדילוגים בגודל skip. לדוגמא, אם
begin=53, end=89, skip=10 אז יודפס: 53 63 73 83.
2. (5 נק') פונקציה בשם `countOddDigits` המקבלת כפרמטר מספר שלם ארוך, ומחזירה
את כמות הספרות האי-זוגיות בו. לדוגמא, אם הפרמטר הוא 6778987 אז יוחזר 4.
3. (5 נק') פונקציה בשם `printLatinsOnly` המקבלת כפרמטר מחרוזת, ומדפיסה את התווים
שהם אותיות לטיניות. לדוגמא, אם הפרמטר הוא: "abc12DEF+3?4aa+a" אז יודפס
abcDEFaaaa.

4. (5 נק') פונקציה בשם **onlyLatins** המקבלת כפרמטר מחרוזת. הפונקציה מחזירה 1 אם כל התווים במחרוזת הם אותיות לטיניות קטנות. אחרת, היא מחזירה 0.

5. (6 נק') פונקציה בשם **consistDgits** המקבלת כפרמטר מחרוזת. הפונקציה מחזירה 1 אם המחרוזת מכילה לפחות סיפרה אחת. אחרת, היא מחזירה 0.

6. (7 נק') סידרה נקראת **סידרת פיבונצ'י** אם מתקיימת אחת משתי האפשרויות הבאות:
a. אורך הסידרה הוא לכל היותר 2.

b. אורך הסידרה גדול מ-2, וכל איבר פרט ל-2 הראשונים שווה לסכום של 2 קודמיו.
כתוב פונקציה בשם **isFibSeries** המקבלת כפרמטר מערך של שלמים וגודלו. הפונקציה מחזירה 1 אם אברי המערך הם סידרת פיבונצ'י. אחרת, היא מחזירה 0.
לדוגמא, אם המערך הוא {7,8,15,23,38,61,99,160,259} אז הפונקציה תחזיר 1.

7. (7 נק') כתבו פונקציה רקורסיבית בשם **checkArrays** המקבלת כפרמטרים שני מערכים של מספרים שלמים, את גודלם המשותף ומספר נוסף בשם value. היא מחזירה 1 אם המערך השני מתקבל מהראשון על-ידי כפל של כל האיברים באותו value. אחרת, היא מחזירה 0. למשל עבור מערך הראשון {15,27,81,65}, מערך השני {150,270,810,650}, גודל כל מערך 4 ומספר נוסף value 10, הפונקציה תחזיר 1.

ניתן להשתמש בדרייבר הקיים (פונקציה הראשית):

```
int main() {
    char str1[] = "abc12DEF+3?4aa+a", str2[] = "abcaaaa";
    int ar1[] = { 7,8,15,23,38,61,99,160,259 }, ar2[] = {7,8,16,23,38};
    int arr3[]={14,16,32,46,76};
    printNumers(53, 89, 10); //53 63 73 83
    printf("\n%d\n", countOddDigits(6778987)); //4
    printLatinsOnly(str1); //abcDEFaaa
    printf("\n%d %d\n", onlyLatins(str1), onlyLatins(str2)); //0 1
    printf("%d %d\n", consistDgits(str1), consistDgits(str2)); //1 0
    printf("%d %d\n", isFibSeries(ar1, 9), isFibSeries(ar2, 5)); //1 0
    printf("%d %d\n", checkArrays(ar2, ar3, 5, 2),); //1
    return 0;
}
```

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

א. נתונות ההגדרות הבאות :

מבנה *Course* המכיל מידע על קורס שלמד סטודנט : שם הקורס והציון:

```
typedef struct {
    char courseName[30] ;
    int grade;
} Course;
```

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

```
typedef struct {  
    char studentName[30] ;  
    Course * course_arr ;  
    int size;  
} Student ;
```

ניתן להניח כי האורך של שם של סטודנט או של קורס אינו עולה על 29 תווים. שמות אלו יכולים להכיל רווחים (יותר ממילה אחת בכל שם).

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 המסך 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 המסך נתונים עבור סטודנט אחד, כולל נתונים על הקורסים של הסטודנט.

Student* Build(int* pSize)
(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 הסטודנטים, 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 בדרייבר (main) הקיים :

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

חלק ג' – קבצים – מענה בקובץ part3.c – 20 נק'

קיים קובץ המכיל רצף של ציונים. כל ציון הינו מספר שלם (**integer**). הציונים מופרדים זה מזה ע"י רווח בודד.

כתבו פונקציה בשם AvgFromFile המקבלת מחרוזת המייצגת את שם הקובץ. הפונקציה מחזירה את הציון הממוצע (ללא התייחסות לערכים שלילים שבקובץ).
לדוגמה , הפונקציה מקבלת את שם הקובץ grade.txt אשר מכיל :

20 45 -56 90 -3 60 100 56

הפונקציה מחזירה :

The average grade : 61.83

כתבו פונקציה ראשית הקולטת מהמשתמש את שם הקובץ , קוראת לפונקציה ומדפיסה את הציון הממוצע. אם הקובץ לא קיים , יש להדפיס הודעה המתאימה ולסיים תוכנית , אם הפונקציה לא מצאה אף ערך מתאים (כל הערכים הם שליליים) יש להדפיס הודעה מתאימה .
הערה: לצורך הבדיקה יש ליצור קובץ ציונים בעזרת ה-visual studio , אבל אין להגיש אותו .

בהצלחה ☺