

מבחן סדנה מתקדמת בתכנות-61108

ד"ר מרק קורנבליט, ד"ר לאוניד קוגל, מר טוביה דב רוזנברג, מר דוד אלגרלי

סמסטר ב', מועד א', תשפ"א

תאריך: 15.06.21

הוראות:

- משך המבחן 3 שעות.
- אין להשתמש בחומרי עזר, פרט לדף המצורף לשאלון.

חלק 1. בחלק זה יש לענות על שתי שאלות מתוך שלוש.

שאלה 1 (40 נק')

נתונה הגדרה הבאה של הטיפוס name אשר מהווה שם פרטי ושם משפחה של בן-אדם:

```
typedef struct  
{  
    char first_name[30];  
    last_name[30];  
} name;
```

כתוב פונקציה אשר מקבלת מערך איברים מסוג **name** וגודלו.
על הפונקציה לבדוק האם איברי המערך ממוינים בסדר לקסיקוגרפי עולה לפי השמות (שדה **first_name**) או לפי שמות המשפחה (שדה **last_name**). אם המערך ממוין לפי השמות, הפונקציה תחזיר **1** ואם לפי שמות המשפחה – הפונקציה תחזיר **2**. אם איברי המערך לא ממוינים בכלל, הפונקציה תחזיר **0**.

דוגמה: עבור המערך הבא

- 0 Mike Asher
- 1 David Bern
- 2 Joseph Borgman
- 3 Alon Kapulski
- 4 Ilan Mizrachi
- 5 Oren Nodel
- 6 Abraham Rosen

הפונקציה תחזיר **2**.



מכון טכנולוגי חולון
Holon Institute of Technology

```
int ordered_name_array(name* name_arr, int n)
{
    int i;
    for (i = 1; i < n; i++)
        if (strcmp(name_arr[i].first_name, name_arr[i - 1].first_name) < 0)
        {
            for (i = 1; i < n; i++)
                if (strcmp(name_arr[i].last_name, name_arr[i - 1].last_name) < 0)
                    return 0;
            return 2;
        }
    return 1;
}
```

גולומב 52, ת.ד. 305, חולון 5810201
טלפקס: 03-5026528

52 Golomb St., Holon 5810201 Israel

www.hit.ac.il Tel/Fax: 972-3-502-6528

הפקולטה למדעים
המחלקה למדעי המחשב

Faculty of Sciences
Department of Computer Science

שאלה 2 (40 נק')

נתונה הגדרה הבאה של הטיפוס:

```
typedef struct item {  
    int data;  
    struct item *next;  
} Item;
```

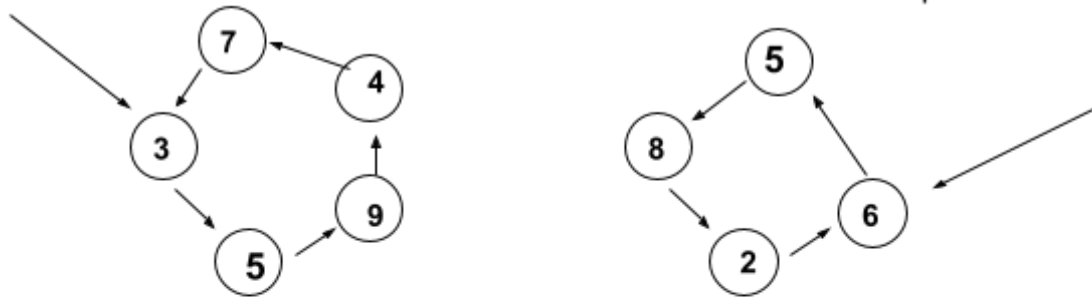
כתוב פונקציה אשר מקבלת שתי רשימות מקושרות **מעגליות** (רשימה מעגלית מועברת לפונקציה כמצביע לאחד מהאיברים שלה) המורכבות מאיברים מסוג **Item**. הרשימות מכילות מספרים שונים כאשר ישנו רק מספר אחד המופיע בשתי הרשימות.

על הפונקציה לאחד את הרשימות לרשימה מעגלית אחת כך שהמספר המשותף יופיע פעם אחת ויחצוץ בין שתי הרשימות המקוריות, שתשמורנה על סידורן הפנימי לאחר האיחוד. לא משנה מאיזו רשימה מקורית יהיו איברים "אחרי" המספר המשותף ברשימה המאוחדת ומאיזו רשימה יהיו "לפניו". גם לא משנה איזה איבר המכיל את המספר המשותף ייכלל ברשימה המאוחדת. הפונקציה תחזיר את כתובת האיבר המכיל את המספר המשותף ברשימה המאוחדת.

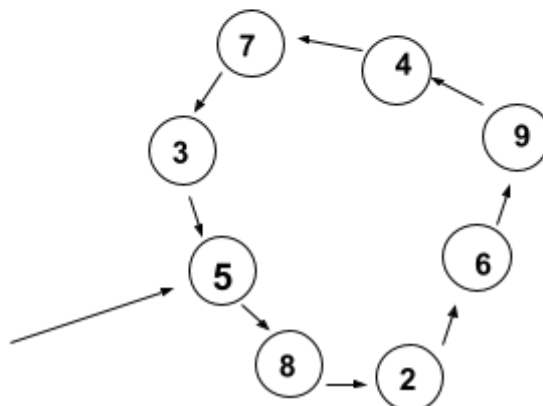
שים לב. הפונקציה לא תקצה זיכרון לרשימה החדשה אלא תעדכן שדה **next** באיברים המתאימים של הרשימות המקוריות.

דוגמא:

רשימות מקוריות:



רשימה מאוחדת:



המספר המשותף הוא 5.



מכון טכנולוגי חולון Holon Institute of Technology

שים לב. כל אחת מרשימות מקוריות יכולה להיות מורכבת גם רק מאיבר אחד (שדה **next** שלו מצביע לאיבר עצמו). במקרה זה האיבר היחיד מכיל את המספר המשותף.

```
/* Auxiliary function. Search of common items.
The function gives addresses of items preceding common ones */
void addresses_of_items_preceding_common(Item *L1, Item *L2, Item **p1, Item **p2){
    Item *prev1 = L1, *prev2, *curr1 = prev1->next, *curr2;
    while (1) //external loop: scan first list
    {
        prev2 = L2;
        curr2 = prev2->next;
        do //internal loop: scan second list
        {
            if (curr2->data != curr1->data)
            {
                prev2 = curr2;
                curr2 = curr2->next;
            }
            else //common items are found
            {
                *p1 = prev1;
                *p2 = prev2;
                return;
            }
        } while (curr2 != L2->next);
        prev1 = curr1;
        curr1 = curr1->next;
    } //while (1)
}

/* Special case: list L1 has one item */
Item *one_item_list(Item *L1, Item *L2){
    int common = L1->data;
    free(L1);
    while (L2->data != common) //search of common item in L2
        L2 = L2->next;
    return L2;
}

Item *cyclic_lists_combining(Item *L1, Item *L2){
    Item *p1, *p2, *common;
    if (L1->next == L1) //L1 has one item
        return one_item_list(L1, L2);
    if (L2->next == L2) //L2 has one item
        return one_item_list(L2, L1);

    //p1 and p2 are addresses of items preceding common ones
    addresses_of_items_preceding_common(L1, L2, &p1, &p2);

    /* Combining of L1 and L2*/
    common = p1->next;
    p1->next = p2->next->next;
```

```
free(p2->next);
p2->next = common;
return common;
}
```

שאלה 3 (40 נק')

כתוב פונקציה אשר מקבלת מחרוזת **fileName** המהווה שם של **קובץ טקסט**. הקובץ שומר תוכן של מערך דו-ממדי של מספרים שלמים שבו כל שורה מורכבת מרצפים של מספרים זהים בדרך חסכונית.

שורה מס' *i* בקובץ מתאימה לשורה מס' *i* של המערך (שורות הקובץ ממוספרות מ-0). כל רצף של מספרים זהים בשורת המערך (מתחילתה עד סופה) מוצג בשורת הקובץ (מתחילתה עד סופה בהתאם) כזוג המספרים: ערך האיברים ברצף וכמות האיברים ברצף. המספרים הסמוכים בכל שורה מופרדים ברווח אחד. התו של סוף השורה מופיע מיד אחרי המספר האחרון בכל שורה, כולל השורה האחרונה.

על הפונקציה לשחזר מערך דו-ממדי דינאמי **A**. יש למלא את **A** בהתאם לתוכן הקובץ. בנוסף הפונקציה תיצור מערך דינאמי **S** המורכב מגדלי שורות של **A** כך שאיבר מס' *i* ב-**S** יהיה שווה לגודל שורה מס' *i* ב-**A**.

הפונקציה תחזיר את **A** ותעביר (by reference) את **S** ואת מספר השורות ב-**A**.

ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

דוגמא:

		<u>תוכן הקובץ</u>
9	4 4 7 7 7 7 3 3 3	4 2 7 4 3 3
8	2 2 2 2 2 1 1	2 6 1 2
4	9 12 12 12	9 1 12 3
9	3 3 18 18 18 18 18 0 0	3 2 18 5 0 2

הפונקציה תעביר 4 כמספר השורות ב-**A**.

```
int **file_to_sequences_in_rows(char* filename, int **S, int *rows)
{
    char ch;
    int **A, num, size, start_row, i, j, k;
    FILE* fptr = fopen(filename, "r");
    if (!fptr)
        exit(1);

    /* Estimation of the number of rows */
    *rows = 0;
    while ((ch = fgetc(fptr)) != EOF)
        if (ch == '\n')
            (*rows)++;

    A = (int**)calloc(*rows, sizeof(int*)); //pointer array allocation
    *S = (int*)calloc(*rows, sizeof(int)); //array of row sizes allocation

    rewind(fptr);
    for (i = 0; i < *rows; i++)
    {
        start_row = ftell(fptr);

        /* Estimation of row size */
        do
        {
            fscanf(fptr, "%d%d", &num, &size);
            (*S)[i] += size;
        } while ((ch = fgetc(fptr)) != '\n');

        A[i] = (int*)calloc((*S)[i], sizeof(int)); //row allocation

        /* File row -> array row */
        fseek(fptr, start_row, SEEK_SET);
        k = 0;
        do
        {
            fscanf(fptr, "%d%d", &num, &size);
            for (j = 1; j <= size; j++, k++)
                A[i][k] = num;
        } while ((ch = fgetc(fptr)) != '\n');
    } //for i

    fclose(fptr);
    return A;
}
```

חלק 2.

שאלה 4 (20 נק')

בחר את התשובה הנכונה ונמק בכל אחד מהסעיפים הבאים (חובה לרשום את הבחירה והנימוק במחברת):

1. מה מאותחל כברירת המחדל?

- (1) משתנה מסוג int שמוגדר בתוך פונקציה, למשל `int x`;
- (2) משתנה מסוג פוינטר ל-int שמוגדר בתוך פונקציה, למשל `int *p`;
- (3) קבועים של טיפוס `enum`
- (4) משתנה בקרה בלולאת for אשר לא מאותחל במפורש בחלק הראשון של הלולאה, למשל `for(; i<n; i++)`
- (5) איברי מערך דינאמי שהוקצה באמצעות פונקציית `malloc`
- (6) אף תשובה אינה נכונה

ערכים של הקבועים של `enum` נקבעים אוטומטית, מתחילים מ-0 וגדלים ב-1.

2. מה קורה בצד שמאל של מספר int חיובי לאחר ביצוע פעולת הזזת סיביות ימינה בתוכו?

- (1) מצד שמאל המספר מתמלא ב-0-ים
- (2) מצד שמאל המספר מתמלא ב-1-ים
- (3) מצד שמאל המספר מתמלא בסיביות שהלכו לאיבוד מצד ימין
- (4) שיטת מילוי המספר מצד שמאל תלויה במערכת
- (5) אף תשובה אינה נכונה

לפי כללי שפת C.

3. איזה מערך מועבר לפונקציה `by value` (ז"א הפונקציה מקבלת גישה להעתק של תוכנו אבל לא לתוכן המקורי שלו)?

- (1) מערך סטטי חד-ממדי
- (2) מערך דינאמי חד-ממדי
- (3) מערך סטטי דו-ממדי
- (4) מערך דינאמי דו-ממדי
- (5) מערך סטטי אשר מהווה שדה של המבנה כאשר המבנה מועבר לפונקציה `by value`
- (6) אף תשובה אינה נכונה

מערך עצמו תמיד מועבר `by reference` כי הפונקציה מקבלת לא מערך עצמו אלא כתובתו. אבל מבנה מועבר `by value` כך שכל השדות שלו מועתקים כולל שדות שהם מערכים.

4. נתונה הפונקציה הבאה:

```
int *fun() {  
    static int x;  
    x++;  
    return &x;  
}
```

מהי הטענה הנכונה?

- (1) הפונקציה לא תקינה רק מכיוון שהיא מחזירה כתובת של משתנה אשר לא קיים מעבר לפונקציה
- (2) הפונקציה לא תקינה רק מכיוון שניגשים בתוכה למשתנה אשר לא מאוחל
- (3) הפונקציה לא תקינה גם מכיוון שהיא מחזירה כתובת של משתנה אשר לא קיים מעבר לפונקציה וגם מכיוון שניגשים בתוכה למשתנה אשר לא מאוחל
- (4) הפונקציה תקינה
- (5) אף טענה אינה נכונה

משתנה static נשמר בזיכרון גם אחרי יציאה מהפונקציה. משתנה static ללא אתחול מפורש מאוחל ב-0.

חלק 3.

שאלה 5 (בנוס – 10 נק')

מהו הפלט של התוכנית הבאה?

```
#include <stdio.h>  
  
void main()  
{  
    printf("%d\n", sizeof"|0\0/00\0/0|");  
}
```

יש לנמק את התשובה.

11



מכון טכנולוגי חולון
Holon Institute of Technology

בטקסט שבתוך המירכאות יש 12 סימנים. ביניהם פעמיים מופיעה סדרת '0' אשר מהווה תו אחד. לכן מופיעים פה 10 תווים. בסוף כל מחרוזת קבועה יש עוד תו אפסי ('0'). לכן סה"כ המערך מורכב מ-11 תווים. אופרטור sizeof מחזיר את גודל האופרנד בבתים. כל תו תופס בית אחד. לכן הפלט יהיה 11.