



מכללת אורט כפר-סבא

## מבני נתונים ויעילות אלגוריתמים

### תרגיל מס' 10

פתרו את השאלות הבאות. יש לסיים את התרגיל עד יום ד' (10.12).

#### שאלה 1

כתבו פונקציה המקבלת מצביע לראש רשימה מקושרת, ומחזירה את מקומו של האיבר האחרון ברשימה. מה לדעתכם צריכה הפונקציה להחזיר במקרה והרשימה היא ריקה?

#### שאלה 2

כתבו פונקציה המקבלת מצביע לראש רשימה מקושרת, ובודקת האם הרשימה ממוינת בסדר עולה. הפונקציה תחזיר 1 אם כן, ו-0 אם לא.

#### שאלה 3

כתבו פונקציה המקבלת מצביע לראש רשימה מקושרת של מספרים שלמים, וכן שני מספרים שלמים  $x$  ו- $y$ . הפונקציה תחזיר 1 אם הערך  $x$  מופיע ברשימה מיד לפני מופע של  $y$  או אם הערך  $y$  מופיע ברשימה מיד לפני מופע של  $x$ , אחרת תחזיר 0.

#### שאלה 4

אם נסמן ב- $n$  את מספר האיברים ברשימה המקושרת, חשבו מהי סיבוכיות זמן הריצה של הפונקציות שכתבתם בשאלות 1 - 3.

#### שאלה 5

א. האם קטע הקוד שראינו בשיעור, המכניס איבר לרשימה ממוינת, יעבוד גם במקרה והרשימה ריקה? ובמקרה והיא מכילה איבר אחד בלבד? ובמקרה והאיבר החדש קטן או גדול מכל איברי הרשימה? שכתבו אותו, במידת הצורך, כך שיטפל גם במקרים אלו בצורה מוצלחת.  
ב. האם קטע הקוד שראינו בשיעור, ההופך את סדר איברי הרשימה, יעבוד גם במקרה והרשימה ריקה? ובמקרה והרשימה מכילה איבר אחד בלבד? שני איברים? שלושה איברים? שכתבו אותו, במידת הצורך, כך שיטפל גם במקרים אלו בצורה מוצלחת.



## שאלה 6

כתבו פונקציה שכותרתה:

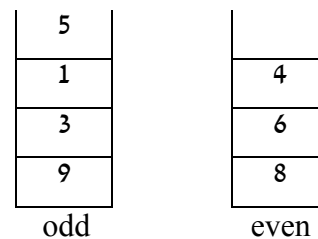
```
void foo (node *list, stack *even, stack *odd);
```

הפונקציה מקבלת מצביע list לראש רשימה מקושרת של שלמים, ובונה שתי מחסניות – מחסנית בשם even המכילה את כל האיברים הזוגיים שברשימה המקושרת, ומחסנית בשם odd המכילה את כל האיברים האי-זוגיים שברשימה המקושרת, לפי הסדר בו הם הופיעו ברשימה.

לדוגמא, עבור הרשימה המקושרת הבאה:



הפונקציה תבנה את שתי המחסניות הבאות (שימו לב לסדר האיברים!):



## שאלה 7

- עליכם לכתוב פונקציה המקבלת שתי רשימות מקושרות  $L_1$  ו- $L_2$ , ומוחקת מהרשימה  $L_1$  את כל האיברים המופיעים ברשימה  $L_2$ . היעזרו בפונקציה `list_search` שראינו בשיעור.
- נסמן ב- $n_1$  את מספר האיברים ברשימה  $L_1$ , ונסמן ב- $n_2$  את מספר האיברים ברשימה  $L_2$ . מהי סיבוכיות זמן הריצה של הפונקציה שכתבתם בסעיף א'?

## שאלה 8

כתבו פונקציה המקבלת מצביע לראש רשימה מקושרת ובודקת האם כל איבר מופיע בה פעם אחת בלבד. הפונקציה תחזיר 1 אם אין איברים זהים, ו-0 אחרת.

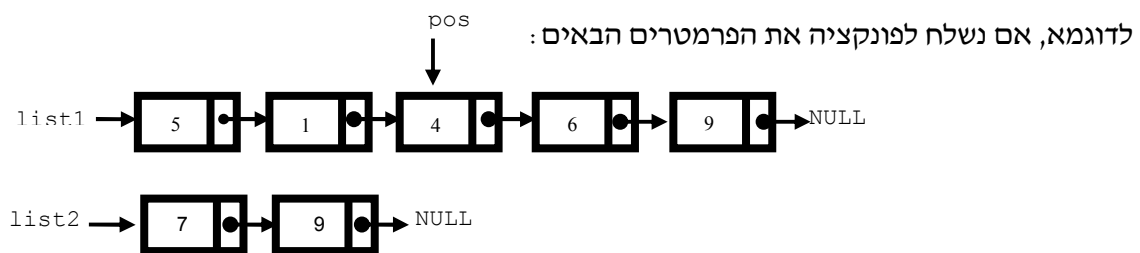


## שאלה 9

כתבו פונקציה שכותרתה:

```
void list_insert_list (node *list1, node *pos, node* list2)
```

הפונקציה מקבלת את list1 שהוא מצביע לראש חוליה ראשונה ברשימה מקושרת, את pos שהוא מצביע לחוליה כלשהי בתוך רשימה מקושרת זו, ואת list2 שהוא מצביע לראש רשימה מקושרת אחרת. הניחו ששתי הרשימות אינן ריקות. הפונקציה תכניס את איברי הרשימה המקושרת list2 לתוך הרשימה list1, אחרי המקום ש-pos מצביע אליו.



אז המצביע list1 יצביע בסיום ריצת הפונקציה לרשימה הבאה (המצביעים pos ו-list2 יכולים להצביע לחוליה כלשהי):



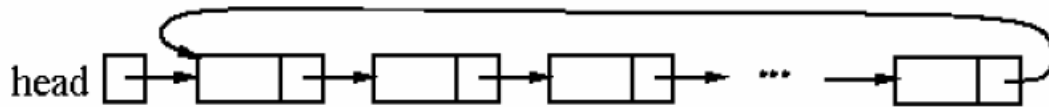
## שאלה 10

בנו יחידת ספרייה לטיפול במחסנית (קובץ ממשק – stack.h וקובץ מימוש – stack.c) אשר מממשת את טיפוס הנתונים המופשט (טני"מ) 'מחסנית' באמצעות רשימה מקושרת.



## שאלה 11

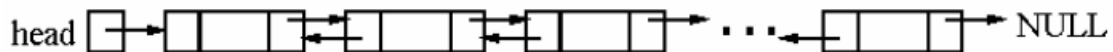
נגדיר רשימה מקושרת מעגלית (circular linked list) בתור רשימה מקושרת שבה שדה ה-`next` של החוליה האחרונה מצביע על החוליה הראשונה, במקום על `NULL`:



כתבו פונקציה המקבלת מצביע לראש רשימה מקושרת רגילה (לא מעגלית) שאינה ריקה, והופכת אותה לרשימה מקושרת מעגלית.

## שאלה 12

נגדיר רשימה מקושרת דו-כיוונית (doubly linked list או bi-directional linked list) בתור רשימה מקושרת שבה בכל חוליה יש לא רק מצביע `next` לחוליה הבאה, אלא גם מצביע `prev` לחוליה הקודמת:



הניחו שבנוסף להגדרת ה-`node` שראינו בכיתה, הוגדר גם:

```
typedef struct double_node_type {
    int info;
    struct double_node_type *next;
    struct double_node_type *prev;
} double_node;
```

כתבו פונקציה המקבלת מצביע לראש רשימה מקושרת חד-כיוונית, ובונה רשימה מקושרת דו-כיוונית המכילה את איברי הרשימה המקושרת החד-כיוונית, באותו הסדר.

