

מבחן מבני נתונים

ענה על כל השאלות

1. (30 נק') נתונה ספרייה הכוללת הגדרות ופונקציות לטיפול במחסנית (Stack), ובה ההגדרות שלהלן:

```
#define STACK_MAX_SIZE 100 // גודל מקסימלי של מחסנית
struct stack {
    int top;
    int data[STACK_MAX_SIZE];
};
```

הפונקציות:

<code>void init(struct stack s)</code>	פונקציה זו מאתחלת את המחסנית s להיות ריקה
<code>void push(struct stack s, int x)</code>	פונקציה זו מוסיפה את האיבר x למחסנית s הנחה: המחסנית s מאותחלת ואינה מלאה
<code>int pop(struct stack s)</code>	פונקציה זו שולפת את האיבר שבראש המחסנית s ומחזירה את ערכו הנחה: המחסנית s מאותחלת ואינה ריקה
<code>int top(struct stack s)</code>	פונקציה זו מחזירה את ערכו של האיבר שבראש המחסנית s מבלי להוציאו מהמחסנית הנחה: המחסנית s מאותחלת ואינה ריקה
<code>int isEmpty(struct stack s)</code>	פונקציה זו מחזירה את הערך 1 אם המחסנית s ריקה, אחרת – היא מחזירה את הערך 0 הנחה: המחסנית s מאותחלת

הנח שהפונקציות האלו כתובות וניתן להשתמש בהן בכל הסעיפים הבאים בלי לכתוב אותן מחדש.

כמו כן, בעבור כל סעיף תוכל להשתמש בכל פונקציה שמומשה בסעיפים שלפניו.

נתונה ההגדרה הבאה:

```
struct stack s;
```

ענה על כל הסעיפים א'-ו':

א. (10 נק') לפניך פונקציה בשם `returnCount` המקבלת מחסנית `s`, ומספר `x`, הפונקציה מחזירה את האיבר `x` מתחילת המחסנית – מבלי להוציאו מהמחסנית. בסיום הפונקציה, המחסנית `s` נשארת ללא שינוי.

דוגמה: עבור המחסנית `s` שלהלן:

1
2
4
5

והמספר `x=3`, הפונקציה תחזיר את המספר 4 שהוא האיבר השלישי במחסנית.

בפונקציה חסרים **ארבעה** ביטויים, המסומנים במספרים בין סוגריים עגולים. רשום ליד כל מספר את הביטוי החסר שהוא מייצג.

```
int returnCount(struct stack s, int count) {
    int total = 0;
    struct stack temp;
    init(temp);
    while (!isEmpty(s))
    {
        push(temp, pop(s));
        (1) _____;
    }
    int i = 1;
    for (; i <= (2) _____; i++)
        push(s, pop(temp));
    int ret = (3) _____;
    push(s, ret);
    (4) _____;
    for (; i <= total; i++)
        push(s, pop(temp));
    return ret;
}
```

ב. (5 נק') מהי סיבוכיות זמן הריצה של האלגוריתם המוצג בסעיף א', בהנחה שמספר האיברים במחסנית `s` הוא n ?

1. $O(n^2)$
2. $O(\log(n))$
3. $O(n)$
4. $O(1)$

ג. (10 נק') לפניך פונקציה בשם palindrom המקבלת מחסנית s המכילה מספרים, ומספר שלם total המייצג את מספר האיברים המחסנית. הפונקציה מחזירה 1 אם המחסנית מייצגת פלינדרום. אם לא, הפונקציה מחזירה 0. בסיום הפונקציה, המחסנית s יכולה להשתנות. למשל, עבור המחסנית s להלן:

1
5
-2
3
-2
5
1

הפונקציה תחזיר 1 שכן המחסנית מייצגת פלינדרום שהוא 1 5 -2 3 -2 5 1.

אבל עבור המחסנית:

1
5
-2
1

הפונקציה תחזיר 0, שכן הביטוי המיוצג 1 5 -2 1, אינו פלינדרום.

בפונקציה חסרים **שלושה** ביטויים המסומנים במספרים בין סוגריים עגולים. רשום ליד כל מספר את הביטוי החסר שהוא מייצג.

```
int palindrom(struct stack s,int total) {
    int i = 0;
    for (;i<total/2; i++)
    {
        int a = (1) _____;
        int b = (2) _____;
        if ((3) _____)
            return 0;
        pop(s);
    }
    return 1;
}
```

ד. (5 נק') מהי סיבוכיות זמן הריצה של האלגוריתם המוצג בסעיף א', בהנחה שמספר האיברים במחסנית s הוא n?

1. $O(n^2)$
2. $O(\log(n))$
3. $O(n)$
4. $O(1)$

2. (30 נק') נתונה ספרייה הכוללת הגדרות ופונקציות לטיפול בתור (Queue), ובה ההגדרות שלהלן:

```
#define QUEUE_MAX_SIZE 100 //גודל מקסימלי של תור
struct queue {
    int head, tail;
    int data[QUEUE_MAX_SIZE];
};
```

הפונקציות:

<code>void init(struct queue q)</code>	פונקציה זו מאתחלת את התור q להיות ריק
<code>void enqueue(struct queue q, int x)</code>	פונקציה זו מוסיפה את האיבר x לתור q הנחה: התור q מאותחל ואינו ריק
<code>int dequeue(struct queue q)</code>	פונקציה זו שולפת את האיבר שבראש התור q ומחזירה את ערכו הנחה: התור q מאותחל ואינו ריק
<code>int peek(struct queue q)</code>	פונקציה זו מחזירה את ערכו של האיבר שבראש התור q מבלי להוציאו מהתור הנחה: התור q מאותחל ואינו ריק
<code>int isEmpty(struct queue q)</code>	פונקציה זו מחזירה את הערך 1 אם התור q ריק, אחרת – היא מחזירה את הערך 0 הנחה: התור q מאותחל

הנח שהפונקציות האלו כתובות וניתן להשתמש בהן בכל הסעיפים הבאים בלי לכתוב אותן מחדש.

כמו כן, בעבור כל סעיף תוכל להשתמש בכל פונקציה שמומשה בסעיפים שלפניו.

`struct queue q;`

נתונה ההגדרה הבאה:

ענה על כל הסעיפים א'-ד':

א. (10 נק') לפניך פונקציה בשם **removeDouble** המקבלת תור שמכיל מספרים שלמים (חיוביים ושליילים). הפונקציה מקבלת תור q ומוציאה איברים כפולים מהתור. בסיום הפונקציה האיברים ב- q נשאר **באותו הסדר** שהיו לפני הקריאה לפונקציה, למעט האיברים הכפולים, שלא נמצאים בתור.

למשל עבור התור לפני הקריאה לפונקציה: (התור משמאל לימין)

5	5	3	1	7	-2	0	3	5	7
---	---	---	---	---	----	---	---	---	---

לאחר ביצוע הפונקציה **removeDouble**:

5	3	1	7	-2	0
---	---	---	---	----	---

שים לב שנמחקו האיברים המאוחרים יותר.

בפונקציה חסרים **ארבעה** ביטויים המסומנים במספרים בין סוגריים עגולים. רשום ליד כל מספר את הביטוי החסר שהוא מייצג.

```
void removeDouble(struct queue q) {
    struct queue tmp1,tmp2;
    init(tmp1); init(tmp2);
    while (!isEmpty(q))
    {
        int x = (1) _____;
        while (!isEmpty(q))
        {
            int y = (2) _____;
            if (x != y)
                (3) _____;
        }
        (4) _____;
        while (!isEmpty(tmp1))
            enqueue(q, dequeue(tmp1));
    }
    while (!isEmpty(tmp2))
        enqueue(q, dequeue(tmp2));
}
```

ב. (5 נק') מהי סיבוכיות זמן הריצה של **removeDouble**, בהנחה שמספר האיברים בתור q הוא n ?

1. $O(n^2)$
2. $O(\log n)$
3. $O(1)$
4. $O(n)$

ג. (10 נק') לפניך פונקציה `insertQueue` המקבלת תור ממין `q`, כך שהמספר הקטן ביותר נמצא **בראש התור**, ומספר נוסף `x`. הפונקציה מכניסה את המספר `x` למקומו הראוי לו בתור, כך שהתור נשאר ממין.

לדוגמה: עבור התור הבא: (משמאל לימין):

1	2	4	5
---	---	---	---

והמספר $x=3$, התור אחרי ביצוע הפונקציה יראה כך:

1	2	3	4	5
---	---	---	---	---

בפונקציה חסרים **שלושה** ביטויים המסומנים במספרים בין סוגריים עגולים. רשום ליד כל מספר את הביטוי החסר שהוא מייצג.

```
int insertQueue(struct queue q, int x)
{
    struct queue tmp;
    init(tmp);
    while (!isEmpty(q) && (1) _____ <= x)
    {
        (2) _____;
    }
    (3) _____;
    while (!isEmpty(tmp))
        enqueue(q, dequeue(tmp));
}
```

ד. (5 נק') מהי סיבוכיות זמן הריצה של הפונקציה `insertQueue`, בהנחה שמספר האיברים בתור `q` הוא n ?

1. $O(n^2)$
2. $O(\log n)$
3. $O(n)$
4. $O(1)$

3. ענה על כל הסעיפים

א. נתון קטע הקוד הבא

```
void Tr(int n)
{
    int k, x;
    if(n>1)
    {
        x=n;
        while(x>1)
            x=x- n/10;
        k=2;
        while(k<n)
            k=k*2;
        Tr(n-1);
    }
}
```

מהי סיבוכיות זמן הריצה כפונקציה של n ? נמק

ב. $T(n) = 2T(\sqrt{n}) + \log_2 n$ היא פונקציית זמן ריצה של אלגוריתם מסויים הפועל על קלט שגודלו n . מהי סיבוכיות זמן הריצה של האלגוריתם?