



מכללת אורט כפר-סבא

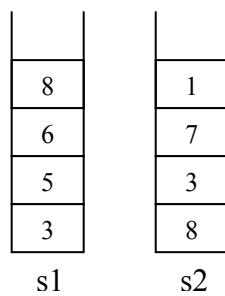
מבני נתונים ויעילות אלגוריתמים

תרגיל מס' 8

פתרו את השאלות הבאות. יש לסיים את התרגיל עד יום ד' (19.11).

שאלה 1

המספר 8653 מאוחסן במחסנית s1 והמספר 1738 מאוחסן במחסנית s2:

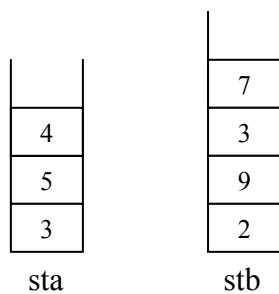


נתונה הפונקציה הבאה:

```
void mystery (stack *s1, stack *s2)
{
    int z;
    stack s3,s4;

    z = 0;
    stack_init(&s3);
    stack_init(&s4);
    while (!stack_empty(*s1))
        stack_push(&s3,stack_pop(s1));
    while (!stack_empty(*s2))
        stack_push(&s4,stack_pop(s2));
    while (!stack_empty(s3) && !stack_empty(s4)) {
        z += stack_pop(&s3) + stack_pop(&s4);
        stack_push(s1,z % 10);
        z /= 10;
    }
    if (z != 0)
        stack_push(s1,z);
}
```

א. מזמנים את הפונקציה באופן הבא: `mystery (&s1, &s2)`. מה יהיה תוכנה של כל



אחת משתי המחסניות s1, s2 לאחר הזימון?

ב. בהינתן 2 המחסניות הבאות, עקבו אחר זימון הפונקציה

`mystery (&sta, &stb)` ורשמו את תוכן לאחר הזימון.

ג. תארו במילים מה מבצעת הפונקציה `mystery`.

שאלה 2

- א. בנו יחידת ספרייה stack.h המטפלת במחסנית של תווים (char), בהסתמך על המימוש שנלמד בכיתה (באמצעות מערך סטטי).
- ב. כתבו פונקציה שכותרתה:

```
int check_char (stack s, char ch)
```

הפונקציה מקבלת מחסנית של תווים וכן תו נוסף, ומחזירה 1 אם התו נמצא במחסנית, ו-0 אם לא. שימו לב שייתכן והמחסנית המתקבלת כפרמטר היא ריקה.

- ג. בדקו את הפונקציה שכתבתם בסעיף ב', בעזרת התכנית הבאה:

```
#include <stdio.h>
#include "stack.h"

int check_char (stack s, char ch)
{
    /* פה ייכתב המימוש שלכם מסעיף ב' */
}

int main()
{
    stack s1;
    char tav;

    stack_init(&s1);
    stack_push(&s1, 'a');
    stack_push(&s1, '(');
    stack_push(&s1, '3');
    scanf ("%c", &tav);
    printf ("%d", check_char(s1,tav));

    return 0;
}
```

שאלה 3

נגדיר "בלוק" במחסנית כרצף של לפחות שני איברים זהים. כתבו פונקציה product_of_blocks המקבלת כפרמטר מחסנית של שלמים, ומחזירה מחסנית חדשה, הבנויה באופן הבא: עבור כל "בלוק" שבמחסנית המתקבלת כפרמטר, יש במחסנית החדשה איבר, שערכו הוא המכפלה של האיבר שב"בלוק" במספר הפעמים שהוא מופיע ב"בלוק". אין חשיבות לסדר האיברים במחסנית המוחזרת.

5
5
5
0
0
0
0
0
3
2
-4
-4

לדוגמא, אם הפונקציה תקבל בתור פרמטר את המחסנית הזו:

-8
0
15

אז היא תחזיר את המחסנית הזו:

שאלה 4

א. כתבו פונקציה שכותרתה:

```
int stack_max (stack s)
```

הפונקציה מקבלת כפרמטר מחסנית של שלמים, ומחזירה את הערך הגדול ביותר המופיע במחסנית. שימו לב שבאחד השיעורים פיתחנו כבר אלגוריתם הפותר בעיה זו, ומה שנדרש מכם בסעיף זה הוא ליישם את האלגוריתם בסביבת העבודה.

ב. כתבו פונקציה שכותרתה:

```
int *build_max_arr (stack a[], int n)
```

הפונקציה מקבלת כפרמטר מערך a המורכב מ-n מחסניות של שלמים, ובונה ומחזירה מערך של שלמים, שאיבריו הם המספרים הגדולים ביותר מכל אחת מהמחסניות. כלומר, האיבר הראשון במערך הוא המס' הגדול ביותר במחסנית הראשונה, האיבר השני במערך הוא המס' הגדול ביותר במחסנית השנייה, וכו'. היעזרו בפונקציה שכתבתם בסעיף א'.

שאלה 5

נתון האלגוריתם הרקורסיבי הבא:

סוף (S)

/* טענת כניסה: האלגוריתם מקבל מחסנית S מאותחלת ולא ריקה */

אם מחסנית-ריקה? (S), אז:

החזר 'אמת'

אחרת:

num ← (S) מחסנית-life

אם ה-3 מחסנית (S) < num, קצו:

החזר סוף (S)

אחרת:

החזר 'שקר'

א. תנו דוגמא למחסנית בת שישה איברים לפחות, שעבורה האלגוריתם יחזיר 'אמת', ודוגמא שעבורה הוא יחזיר 'שקר'.

ב. איזו משימה מבצע האלגוריתם?

ג. האם הדרישה המופיעה בטענת הכניסה של האלגוריתם, לכך שהמחסנית S תהיה לא ריקה, היא הכרחית? הסבירו את תשובתכם.

ד. ממשו את האלגוריתם בסביבת העבודה, ובדקו אותו עם הקלט שהצעתם בסעיף א'.



שאלה 6

כתבו פונקציה המקבלת מחסנית המכילה תווים, ובודקת האם רצף התווים השמור במחסנית הוא פלינדרום (כזכור, רצף ייקרא פלינדרום אם הוא נקרא באותו הצורה משני הכיוונים).

לדוגמא: המחסנית ['a', 'b', 'b', 'a', 'd'] לא מכילה פלינדרום (הניחו שהתו השמאלי ביותר הוא ראש המחסנית), ואילו המחסנית ['x','y','x'] מכילה פלינדרום.

שאלה 7

נתונה הפונקציה הבאה, המקבלת מספר טבעי:

```
int mystery (int num)
{
    int x = 0, y = 1;
    stack s;
    stack_init(&s);
    while (num != 0) {
        stack_push(&s,num%10);
        num /= 10;
    }
    while (!stack_empty(s)) {
        x += y * stack_pop(&s);
        y *= 10;
    }
    return x;
}
```

א. מהו המספר שיוחזר בעקבות הזימון `mystery(2047)`?

ב. כתבו את טענת היציאה של הפונקציה `mystery`.

שאלה 8

ממשו פונקציה שכותרתה:

```
void merge_sort (int a[], int low, int high)
```

הפונקציה מקבלת מערך `a[]` של שלמים, ואינדקסים המקיימים: $0 \leq low \leq high$. היא תמייך באופן רקורסיבי את המערך `a[low...high]`, באמצעות האלגוריתם למיון מיזוג (Merge Sort), אך תעשה זאת בסדר יורד, ולא בסדר עולה כפי שלמדנו בכיתה.



שאלה 9

כתבו פונקציה המקבלת מחרוזת תווים. הפונקציה תדפיס את התווים בסדר בו הם מופיעים במחרוזת, עד שיופיע התו '@'. תו זה לא יודפס, אבל יגרום להיפוך סדר ההדפסה של התווים בינו ובין ה-'@' הבא. הניחו שסדרת הקלטים מכילה מספר זוגי (אולי אפס) של תווי '@'.

@ven@@im re@nd

: לדוגמה, עבור כל אחד מהקלטים הבאים:

ne@m rev@ind

never mind

יתקבל הפלט: nevermind.

ועבור הקלט: se@tp@emb@re@ so@gn@ : september song יתקבל הפלט:

שאלה 10

כתבו פונקציה המבצעת את אותה המשימה כמו בשאלה הקודמת, אולם כעת כתבו אותה תחת ההנחה המגבילה, האומרת שאת כל אחד מהתווים במחרוזת ניתן לקרוא פעם אחת בלבד. כלומר, לא ניתן לרוץ על פני המחרוזת, לחזור לאחור, לרוץ שוב קדימה, לחזור שוב לאחור, וכו'.

הדרכה: ניתן להיעזר במחסנית.

שאלה 11

השתמשו במשפט 2 ומצאו חסמים אסימפטוטיים בעבור נוסחאות הנסיגה האלה:

א. $T(n) = 2T(n/4) + 5$

ב. $T(n) = 4T(n/4) + 1$

ג. $T(n) = 4T(n/2) + 22$

ד. $T(n) = T(n/2) + 7.5$



שאלה 12

בתרגיל מס' 4 במבני נתונים ויעילות אלגוריתמים, בשאלה 13, ראינו את הפונקציה הרקורסיבית:

```
void find_min_max (int a[], int *min, int *max, int n)
```

הפונקציה קיבלה מערך a של מס' שלמים שגודלו n , והחזירה (דרך הפרמטרים min ו- max , המועברים לפי כתובת) את האיבר המינימלי והאיבר המקסימלי במערך.

א. כתבו נוסחת נסיגה $T(n)$ המתארת את זמן הריצה של פונקציה רקורסיבית זו, והשתמשו בשיטת האיטרציה על מנת להגיע לפתרון מפורש (ביטוי סגור, ללא רקורסיה).

ב. להלן מימוש אחר של אותה פונקציה רקורסיבית מתרגיל מס' 4. מימוש זה פותר בדיוק את אותה הבעיה האלגוריתמית, אך עובד בשיטת הפרד-ומשול (divide-and-conquer):

```
void find_min_max (int a[], int *min, int *max, int n)
{
    int min1,min2,max1,max2;

    if (n == 1)
        *min = *max = a[0];
    else {
        find_min_max(a,&min1,&max1,n/2);
        find_min_max(a+n/2,&min2,&max2,n-n/2);
        *min = (min1 < min2) ? min1 : min2;
        *max = (max1 > max2) ? max1 : max2;
    }
}
```

ג. כתבו נוסחת נסיגה $T(n)$ המתארת את זמן הריצה של הפונקציה $find_min_max$, והשתמשו באחד המשפטים שנלמדו בכיתה על מנת לפתור את נוסחת הנסיגה.

ד. האם האלגוריתם מסעיף ב' יעיל יותר אסימפטוטית (בסדר גודל) מאשר האלגוריתם שהכרנו בתרגיל מס' 4? הסבירו את תשובתכם.

```
int getRandomNumber()
{
    return 4; // chosen by fair dice roll.
             // guaranteed to be random.
}
```