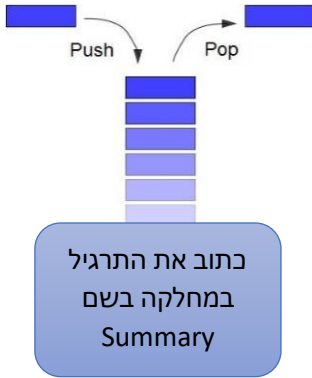


מחסנית – תרגיל סיכום



שימו לב! ♥

1. לפני כל גישה לראש המחסנית (על ידי pop או top) יש לוודא שהמחסנית אינה ריקה באמצעות הפעולה isEmpty
2. כשכותבים פעולה המשתמשת במחסנית יש לשים לב אם התבקשתם להחזיר את מבנה המחסנית למקור בסוף השימוש. במקרה כזה יש לשמור את הנתונים במחסנית זמנית ולהחזירם למחסנית המקורית בסוף בפעולה

1. כתוב פעולה בשם ChangeHalf (Stack<Character> s) המבצעת החלפה של האיברים שבחצי העליון של המחסנית עם האיברים שבחצי התחתון. הנחה: S מכילה מספר זוגי של איברים. ניתן להשתמש במחסניות עזר.

דוגמא:

D	A
E	B
F	C
A	D
B	E
C	F

S אחרי

S לפני

- שם הפעולה stackPick
- אם המחסנית ריקה יוחזר -1
- אין לקלקל את מחסנית

2. נתונה מחסנית S המכילה מספרים שלמים המסודרים באופן הבא: מתחתית המחסנית ועד לאיבר מסוים במחסנית המספרים מסודרים בסדר עולה (האיבר הקטן ביותר נמצא בתחתית המחסנית), ומאותו מקום עד לראש המחסנית המספרים מסודרים בסדר יורד. לדוגמא: 1,3,6,7,5,4,3,2,1. כתוב פעולה המקבלת מחסנית ומחזירה את האיבר בו סדר האיברים התהפך. למשל, עבור המחסנית שבדוגמא יוחזר הערך 7.

putInPlace(Stack<Double> s, double num)

- שם הפעולה mergeStacks
- המחסניות מסוג Integer
- אין לקלקל את מחסניות הקלט

3. כתוב פעולה המקבלת מחסנית המכילה ערכים ממיונים בסדר עולה, ואיבר נוסף. הפעולה תחזיר את המחסנית לאחר שהאיבר הוכנס למקום המתאים בה.

4. כתוב פעולה המקבלת שתי מחסניות ממיוניות בסדר עולה (האיבר הגדול בראש המחסנית), ומחזירה מחסנית חדשה המכילה את כל האיברים של שתי המחסניות המסודרים בסדר יורד (האיבר הקטן בראש).

5. כתוב פעולה המקבלת מחרוזת תווים ומחזירה אמת אם המחרוזת מהצורה xCy, אחרת יוחזר שקר.

xCy- כאשר x- היא מחרוזת תווים המורכבת מהתווים a ו-b.
 y- היא היפוכה (תמונת ראי) של x.
 דוגמא: אם x=abbab אז y=babba
 המחרוזת כולה: abbabcbabba
 (רמז: יש להעזר במחסנית עזר לפתרון התרגיל).

שם הפעולה: isMirror

6. כתוב פעולה המקבלת מחסנית תווים ומחזירה מחסנית חדשה ובה כל רצף של תווים במחסנית שהתקבלה יופיע פעם אחת במחסנית שתוחזר.

המחסנית שתוחזר

המחסנית שהתקבלה

- שם הפעולה zip
- אין לקלקל את מחסניות הקלט

R
G
A
K

R
G
G
A
K
K
K

```

public static int Dolt(Stack<Integer> s1)
{
    Stack<Integer> s2=new Stack<Integer> ();
    int x;
    if (! s1.isEmpty())
        x= s1.pop();
    while (!s1.isEmpty())
    {
        if(x<s1.top())
            s2.push(s1.pop());
        else
        {
            s2.push(x);
            x=s1.pop();
        }
    }
    while(!s2.isEmpty())
        s1.push(s2.pop())
    return x;
}

```

7.

- א. נניח כי האיברים במחסנית s1 הם: 2,5,8,3,0,1,4,9 (המספר 2 נמצא בראש המחסנית). הרץ את הפעולה ורשום מהו הערך המוחזר. רשום את תוכן 2 המחסניות בכל שלב שלאחר שליפה או דחיפה.
- ב. רשום מה מבצעת פעולת Dolt

ג. להלן הפעולה sod המשתמשת בפעולה Dolt. הרץ את הפעולה ורשום מה היא מבצעת.

(הערה: בהרצת הפעולה, אין צורך להריץ את הפעולה Dolt הנמצאת בתוכה בשנית. יש להתייחס אליה כפעולה שאנו יודעים מה היא מבצעת ולקחת את הערך המוחזר ממנה ישירות.)

```
public static void sod(Stack<Integer> s1)
{
    Stack<Integer> s2= new Stack<Integer>();
    while( !s1.isEmpty())
        s2.push(Dolt(s1));
    while( !s2.isEmpty())
        s1.push(s2.pop());
}
```

בהצלחה