© איילת ויץ

בס"ד

תרגול מחסנית

תרגול בסיסי

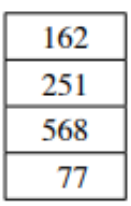
1. כתבי תוכנית ובה:
   1. הגדירי מחסנית ריקה
   2. הוסיפי למחסנית את המספרים 5,8,2
   3. בצעי פעולת הסרה על המחסנית
   4. ציירי את מצב המחסנית הנוכחי
2. כתבי פעולה אשר תקבל מחסנית ותחזיר את האיבר הגדול ביותר במחסנית
3. כתבי פעולה חיצונית המקבלת מחסנית המכילה סוגריים ומחזירה האם מדובר במחסנית מאוזנת.
   מחסנית מאוזנת היא מחסנית שנקראת מלמעלה-למטה ובאופן זהה מלמטה-למעלה.
   לדו', מחסנית מאוזנת: {{[]}}) מחסנית לא מאוזנת: (({}))
4. כתבי פעולה חיצונית המקבלת מחסנית אותיות והופכת את סדר האותיות במחסנית.
   לדו', אם המחסנית שהתקבלה: {h,e,l,l,o} לאחר הפעולה המחסנית תיראה כך: {o,l,l,e,h}
5. כתבי פעולה אשר תקבל מחסנית ותסיר ממנה את האיבר האמצעי. (הניחי כי מס' האיברים אי זוגי)

תרגול מתקדם

1.

א. כתוב פעולה חיצונית isExist בשפת Java או IsExist בשפת C# . הפעולה תקבל מספר num — מטיפוס שלם בין 0 ל- 9 (כולל), ומחסנית stk — מטיפוס שלם. הפעולה תחזיר true אם יש במחסנית מספר שספרת האחדות שלו שווה למספר num , אחרת היא תחזיר false .
הנח שהמספרים במחסנית stk אינם שליליים.

דוגמה: עבור num שווה ל- 8 והמחסנית stk שלפניך:



הפעולה תחזיר true , משום שבמחסנית יש מספר 568 שספרת האחדות שלו היא 8 .

הערה: חובה לשמור על מבנה המחסנית עם סיום הפעולה.

ב. לשם פתרון סעיף זה בלבד תוכל להשתמש בפעולה שלפניך ללא צורך לממש:

תיאור הפעולה	כותרת הפעולה
הפעולה מקבלת מחסנית מטיפוס שלם ומחזירה העתק מדויק של המחסנית בלי לשנות את המחסנית המקורית.	**בשפת Java** — public static Stack <Integer> clone (Stack <Integer> s) **בשפת C#** — public static Stack <int> Clone (Stack <int> s)

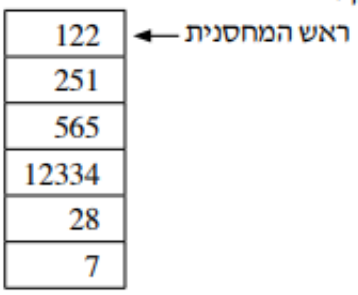
נגדיר: הספרה המשמעותית במספר היא הספרה הכי שמאלית שלו.

לדוגמה הספרה 3 היא הספרה המשמעותית במספר 32 , והספרה 5 היא הספרה המשמעותית במספר 541 .

כתוב פעולה חיצונית allExist בשפת Java או AllExist בשפת C# , המקבלת מחסנית stk שאינה ריקה, מטיפוס שלם. הפעולה תחזיר true אם כל הספרות המשמעותיות במספרים שבמחסנית מופיעות בספרת האחדות במספרים כלשהם במחסנית, אחרת היא תחזיר false .

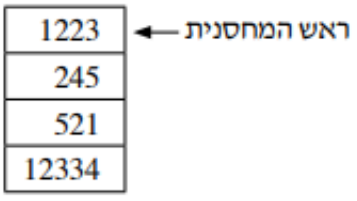
הנח שהמספרים במחסנית stk אינם שליליים.

דוגמה: עבור המחסנית stk שלפניך:



הפעולה תחזיר true כי כל הספרות המשמעותיות במספרים שבמחסנית — 1 , 2 , 5 , 7 — מופיעות בספרת האחדות במספרים שבמחסנית.

דוגמה נוספת: עבור המחסנית stk שלפניך:



הפעולה תחזיר false כי מתוך הספרות המשמעותיות שבמחסנית — 1 , 2 , 5 — הספרה 2 אינה מופיעה בספרת האחדות בשום מספר מן המספרים שבמחסנית.

הערה: אפשר להשתמש בפעולה שכתבת בסעיף א.

.2

נתונה המחלקה **TwoStack** , ולה שתי תכונות:

- numbers – מחסנית מטיפוס שלם
- sums – מחסנית מטיפוס שלם

היחס בין המחסנית sums למחסנית numbers הוא כמפורט להלן:

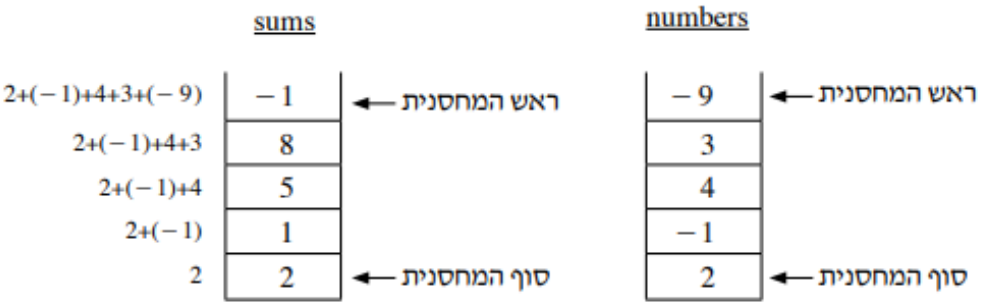
המספר בסוף המחסנית sums שווה למספר בסוף המחסנית numbers ,

המספר השני מסוף המחסנית sums שווה לסכום שני המספרים האחרונים במחסנית numbers ,

המספר השלישי מסוף המחסנית sums שווה לסכום שלושת המספרים האחרונים במחסנית numbers ,

וכן הלאה עד המספר בראש המחסנית sums , השווה לסכום כל המספרים במחסנית numbers .

דוגמה:



א. ממשו את הפעולה הפנימית שלהלן:

Java – public Stack<Integer> getNums (int x)

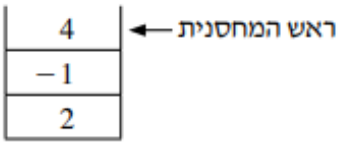
C# – public Stack<int> GetNums (int x)

הפעולה מקבלת מספר x השווה לאחד המספרים במחסנית sums , ומחזירה מחסנית חדשה מטיפוס שלם, שבה מופיעים המספרים מן המחסנית numbers שסכומם שווה למספר x .

הניחו שהמספר x קיים במחסנית sums ומופיע בה רק פעם אחת.

דוגמה:

בעבור המחסניות שבדוגמה לעיל ו־ x = 5 (5 הוא המספר השלישי מלמטה במחסנית sums), הפעולה תחזיר את המחסנית שלהלן:



הערות: – אפשר לשנות את המחסניות של המחלקה.

– אין חשיבות לסדר המספרים במחסנית המוחזרת.

– ההגדרה למחסנית מטיפוס שלם Stack<int>/Stack<Integer> שקולה להגדרה StackInt . אפשר להשתמש בכל אחת מהן, לבחירתכם.

ב. ממשו את הפעולה הפנימית שלהלן:

Java – public void eraseNum (int x)

C# – public void EraseNum (int x)

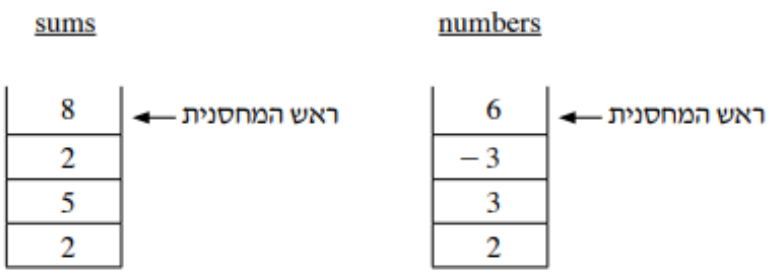
הפעולה מוחקת את המספר x מן המחסנית numbers ומתקנת את המחסנית sums בהתאם.

הניחו שהמספר x קיים במחסנית numbers והוא מופיע בה רק פעם אחת.

הערה: יש לשמור על סדר המספרים שנשארו במחסנית numbers .

דוגמה:

בעבור המחסניות שלפניכם ו־ $x = 3$:



לאחר הרצת הפעולה ייראו המחסניות כך:

