

5. 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:

	<u>sums</u>		<u>numbers</u>	
$2+(-1)+4+3+(-9)$	-1	← ראש המחסנית	-9	← ראש המחסנית
$2+(-1)+4+3$	8		3	
$2+(-1)+4$	5		4	
$2+(-1)$	1		-1	
2	2	← סוף המחסנית	2	← סוף המחסנית

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- $x = 5$ (הוא המספר השלישי מלמטה במחסנית sums), הפעולה תחזיר את המחסנית שלהלן:

4	← ראש המחסנית
-1	
2	

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.

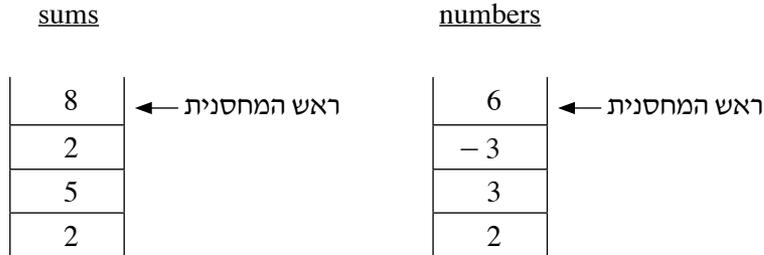
Java – `public void eraseNum (int x)`

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 $x = 3$:



לאחר הרצת הפעולה ייראו המחסניות כך:

