



קרית חינוך "פארק המדע"

בית לערכמים, למצינות וחדשות

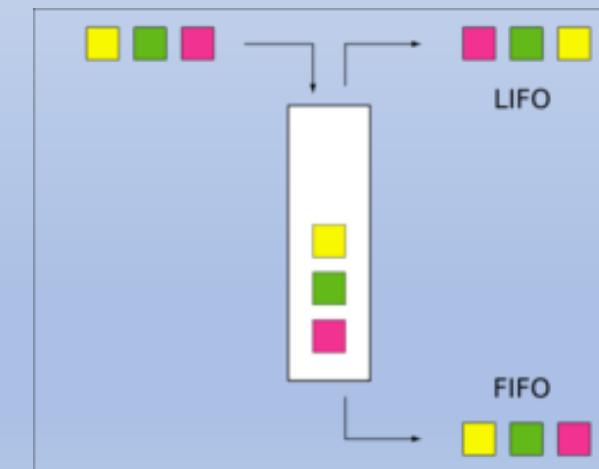
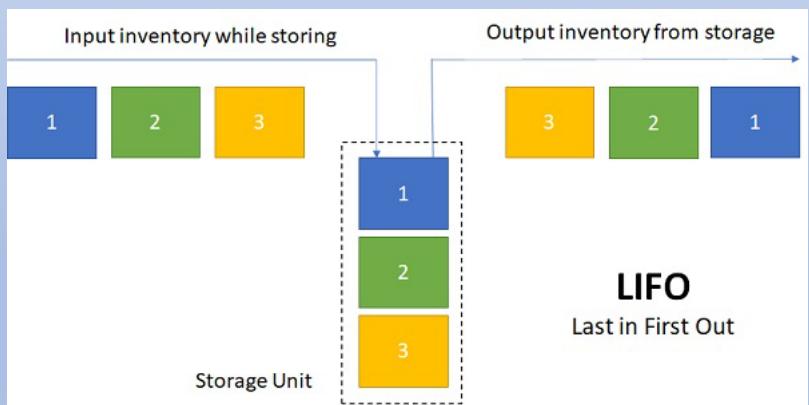


# תור Queue

כיתה יא  
גלעד מרכמן

# טור QUEUE

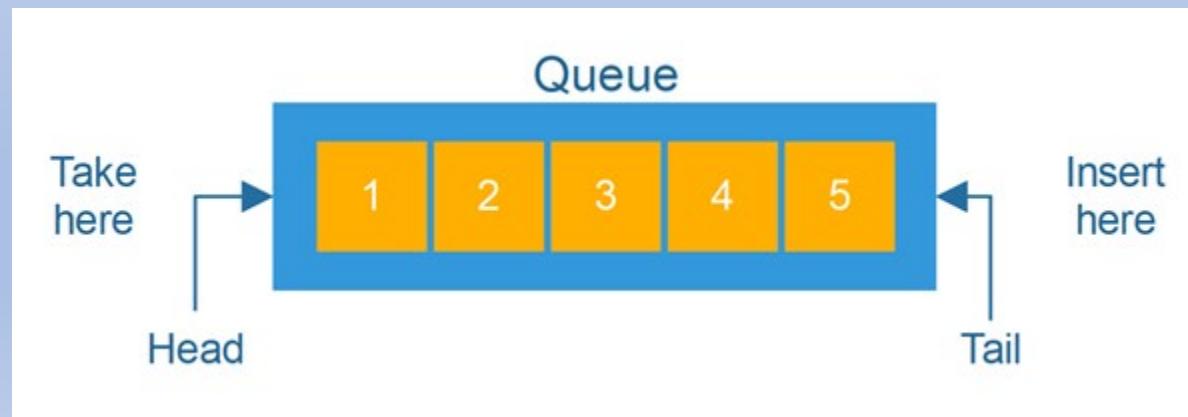
- טור – מבנה נתוניים המקיימים סדרה של נתונים.
- איבר הנכנס למחסנית ישמר בסוף הטור.
- איבר הנכנס מעליו יהיה לאחורי בטור.
- הוצאה איבר תעשה תמיד מראש הטור.





# מיימוש תור גנרי על ידי חוליות

- **מיימוש תור על ידי שרשרת חוליות** מחייב אותנו להחזיק שני מצביעים:
  - Head - מצביע לראש השרשרת – לצורך הוצאת ערכים.
  - Tail – מצביע לסוף השרשרת – לצורך הכנסת הערכים.
- שימוש בשני מצביעים מאפשר לנו לבצע הוספה והסרה של חוליה ב(1)0.
- שימוש בחוליות גנריות מאפשר לבנות תור גנרי.





# המחלקה Queue<T>

```
public class Queue <T>
{
    private Node<T> first;
    private Node<T> last;
    0 references
    public Queue()...
    0 references
    public bool isEmpty ()...
    0 references
    public T Head()...
    0 references
    public void Insert (T value) // ...
    0 references
    public T Remove () // ...
    0 references
    public override string ToString()...
}
```

מוסיף איבר לסוף התור // ...  
מוסיאה ומוחזיר את האיבר הראשון בתור // ...



# המחלקה Queue<T>

```
public class Queue <T>
{
    private Node<T> first;
    private Node<T> last;
    0 references
    public Queue()...
    0 references
    public bool isEmpty ()...
    0 references
    public T Head()...
    0 references
    public void Insert (T value) // מוסיף איבר לסוף תור ...
    0 references
    public T Remove () // מוציא ומחזיר את האיבר הראשון בתור ...
    0 references
    public override string ToString()...
}
```

[30, 15, 20, 6, 3, 7, 4]

- **מחלקה הGeneraty Queue<T> כוללת :**
  - מאפיין Head המצביע לראש המחסנית.
  - **בנייה ריק** – הבונה מחסנית ריקה.
  - **Push** – מוסיף איבר לראש המחסנית.
  - **Pop** – מוציא איבר מראש המחסנית ומחזיר את ערכו.
  - **Top** –מחזיר את ערך האיבר בראש המחסנית (לא משנה את המחסנית).
  - **isEmpty** –מחזיר אמת אם המחסנית ריקה, אחרת שקר.
  - **ToString** –מחזיר מחרוזת עם ערכי המחסנית.  
**בפורמט:**



# תרגיל – ייצירת תור ממערך

- כתוב פעולה חיצונית גנריית מקבלת מערך מסוג `<T>` ומחזירה תור עם ערכי המערך.
- צור והדפו בפעולה הראשית מחסנית של `int` שבנויה ממערך.

```
1 reference
public static Queue<T> CreateQueue <T>(T[] arr)
{
    Queue<T> q = new Queue<T>();
    for (int i=0; i < arr.Length; i++)
    {
        q.Insert(arr[i]);
    }
    return q;
}
```

```
static void Main(string[] args)
{
    Queue<int> q = CreateQueue(new int[] { 3, 7, 2, 12, 34, 1 });
    Console.WriteLine(q);
}
```



# תרגיל – שירשור רשימות

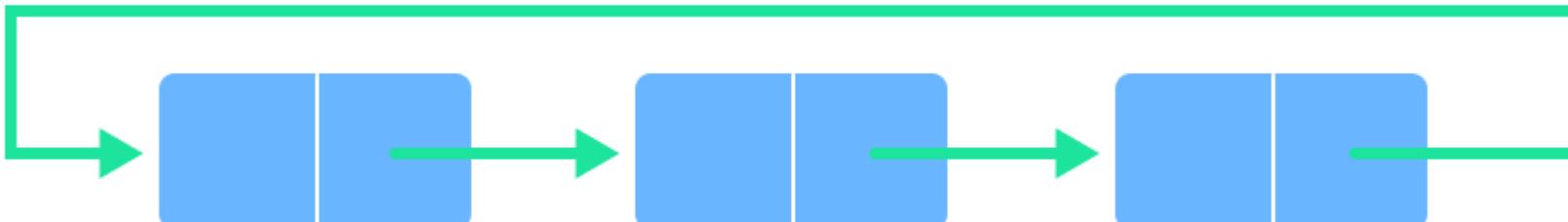
- כתוב פעולה חיצונית גנריית מקבלת שני תורים  $q_1, q_2$  ומשרשרת אותם לטור אחד  $q_1$  (מחברת את  $q_1$  עם  $q_2$ ).

```
3 references
שרשור // Concatenate<T>(Queue<T> q1, Queue<T> q2)
{
    while (!q2.isEmpty())
    {
        q1.Insert(q2.Remove());
    }
}
```



# סרייקת תור

- כתוב פועלה חיצונית גנריית מקבלת תור וערך, ומחזירה אמת אם הערך קיים בתור וSKU אחרה. התור המקורי לא ישתנה.
- נציג שלוש אפשרויות לסריקה של תור ללא שינוי התור המקורי:
  - שימוש בתור נוסף (העתיקת התור בהתחלה או במהלך ביצוע הפעולה).
  - במקרה של ערכים ייחודיים – שמירת הערך בהתאם.
  - הוספת דגל – כשידוע SKU לא קיים בתור.
  - כשידוע מספר הערכים – שימוש בלולאת `for`.





# סריקה באמצעות תור עזר

```
reference
public static bool isExist1<T>(Queue<T> q, T value)
{
    Queue<T> temp = new Queue<T>();
    bool exist = false;
    while (!q.isEmpty())
    {
        T val = q.Remove();
        temp.Insert(val);
        if (val.Equals(value))
        {
            exist = true;
        }
    }
    Concatenate(q, temp);
    return exist;
}
```



# סיקה בתור עם ערכים יהודים

```
1 reference
public static bool isExist2<T>(Queue<T> q, T value)
{
    bool exist = false;
    T flag = q.Head();
    if (flag.Equals(value))
        return true;
    q.Insert(q.Remove());
    while (!q.Head().Equals(flag))
    {
        if (q.Head().Equals(value))
            exist = true;
        q.Insert(q.Remove());
    }
    return exist;
}
```



# סריקה באמצעות דגל

```
1 reference
public static bool isExist3 (Queue<int> q, int value)
{
    bool exist = false;
    int flag = -1;
    q.Insert(-1);
    while (!q.Head().Equals(flag))
    {
        if (q.Head().Equals(value))
            exist = true;
        q.Insert(q.Remove());
    }
    q.Remove();
    return exist;
}
```



# תרגילים – פעולות חיצונית

הפעולה	המהות
CreateQueue <T>	פעולה גנרטיב חיצונית מקבלת מערך של <T> ויוצרת ממנו תור.
Concatenate<T>	שרשור – פעולה גנרטיב מקבלת שני תורים q1,q2 ומשרשרת אותם לתור אחד q (מחברת את q1 עם q2).
isExist1<T>	פעולה גנרטיב מקבלת תור וערך ובודקת אם הערך קיים בתור. התור אינו משתנה. שימוש בתור עזר
isExists2<T>	פעולה גנרטיב מקבלת תור של איברים שונים זה מזה ועריך, ובודקת אם הערך קיים בתור. התור אינו משתנה ואין להשתמש במבנה נתונים נוסף.
isExist3	פעולה מקבלת תור של מספרים חיובים ועריך, ובודקת אם הערך קיים בתור. התור אינו משתנה ואין להשתמש במבנה נתונים נוסף.
Copy	פעולה גנרטיב חיצונית מקבלת תור ומציירה העתק של תור. המחסנית המקורית אינה משתנה.
Reverse<T>	פעולה גנרטיב מקבלת תור והופכת את סדר האברים.
PopMin	פעולה גנרטיב מקבלת תור של מספרים שלמים ומוציאה ומציירה את המספר המינימלי. התור לא ישתנה כמעט הוצאת המספר המינימלי.
Sort	כתבו פעולה מקבלת תור to ומציירה תור חדש הממיין מהמספר הקטן לגדול (המספר הקטן בראש המחסנית). בסיום הפעולה התור המקורי לא משתנה.



# תרגילים – פעולות חיצונית

הפעולה	מהות
Sequence	כתבו פעולה המקבלת תור של מספרים שלמים ומספר שלם $n$ . הפעולה תחזיר <code>true</code> אם מספר החזרות של כל אחד מהמספרים בטוחה $n-1$ הוא כערך המספר עצמו: המספר 1 מופיע פעם אחת, המספר 2 מופיע פעמיים וכך הלאה. אין חשיבות לסדר המספרים. אין להשתמש באלגוריתם למיון. התור לא ישנה. (שאלה 5 בספר).
SortedStack	כתבו פעולה חיצונית המקבלת תור המכיל מספרים טבעיים שונים זה מזה. על הפעולה להחזיר מחסנית ממינית בסדר עולה (האיבר הגדול ביותר בראש המחסנית) הבנייה מכל אברי התור, תוך התחשבות במוגבלות הבאות: (1) בשום שלב אין לשלווף איברים מהחסנית; (2) אין להשתמש במבנה נתונים נוספים למעט התור שהתקבל.
Merge	כתבו פעולה חיצונית המקבלת שני תורים $T_1$ ו- $T_2$ של מספרים ממשיים <code>double</code> וממינים מהקטן לגודל. הפעולה תמזג אותם לתוך $T_1$ כטור יחיד ממויין.
bestStudents	כתבו פעולה המקבלת תור של תלמידים, ומחזירה מחסנית של התלמידים עם הציון הכי גבוה (יכולים להיות מספר תלמידים שקיבלו את הציון הכי גבוה). הערך הוא אובייקט <code>Student</code> הכלול שמו <code>name</code> - <code>grade</code> . הניחו שיש פועלות <code>get</code> , <code>set</code> , <code>get</code> במחלקה ובנאים כפי שתבחרו. סיבוכיות הזמן חייבת להיות $O(n)$ .

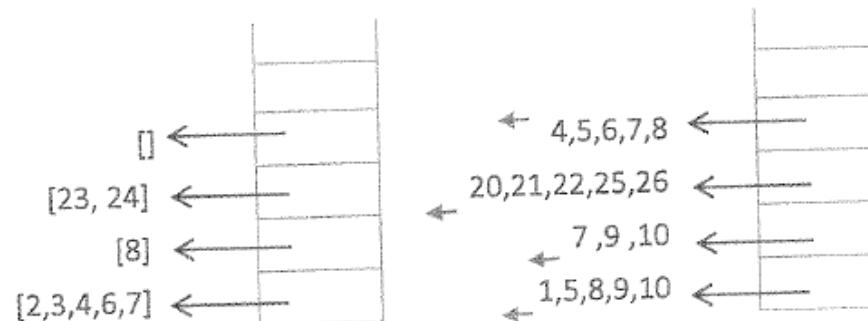


# מחסנית של תורים

## • שאלה 13 בספר

### שאלה 13: מחסנית של תורים – פיתוח ★★☆

נתונה מחסנית SQ שבל איבר בה הוא תור של מספרים שלמים. כל תור ממוקן בסדר עולה כך שהאיבר הראשון בתור הוא הקטן ביותר. לדוגמה: נתונה המחסנית SQ (מימין לשמאל)



LS – המחסנית המתבקשת

SQ – המחסנית המוחזרת

א. ביטבו פעולה חיצונית המתקבלת מחסנית של תורים SQ, ומחדירה מחסנית LS שבל איבר בה הוא שרשרת חוליות. LS היא מחסנית **משלימה** למחסנית SQ כך שבכל שרשרת הממוקמת במחסנית LS נמצאים כל המספרים החסרים בתור הממוקם באותו מקום במחסנית SQ, כדי שייהי רצף של מספרים עוקבים מהקטן ועד הגדול. על המספרים בשרשורת להיות ממוקמים מהקטן לגדול כך שהמספר בתחילת השרשרת הוא הקטן ביותר.

ב. מה סיבוכיות זמן הריצה של הפעולה שכובבתם? נמקו.

- א. פטור במקומות מחסנית יהיה כתוב תור
- ב. במקומות מחסנית יהיה כתוב מערך



# שאלה מבגרות 2021

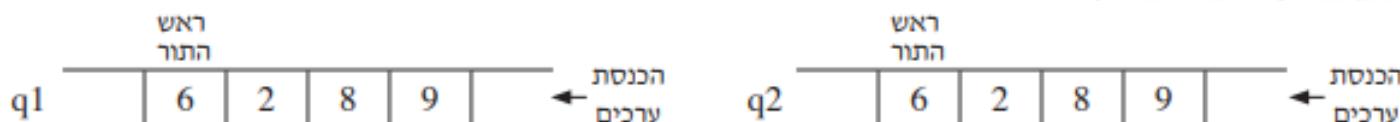
בשאלה זו תוכל להשתמש בפעולה החיצונית שלפניך בלי ממש אותה.

.7

כותרת הפעולה	תיאור הפעולה
public static int size (Queue<Integer> q) – בשפת Java	הפעולה ממחישה את מספר האיברים בתור q בלי לשנות את התור.
public static int Size (Queue<int> q) – בשפת C#	

א. שני תורים,  $q_1$  ו- $q_2$ , יהיו "תורים זהים" אם מספר האיברים בשני התורים זהה, ובשני התורים מופיעים בדיקון אותם ערכים ובאותו הסדר.

דוגמה לשני תורים זהים:



כתב פעלת חיצונית ששמה `isIdentical` בשפת Java או `IsIdentical` בשפת C# המתקבלת שני תורים מטיפוסם של  $q_1$  ו- $q_2$ , וממחישה `true` אם התורים זהים, אחרת היא ממחישה `false`.

הערה: עם סיום הפעולה, חוובה לשמר על מבנה התורים המקורי שהתקבל.

- כתוב בנוסף גם את הפעולה `size`.

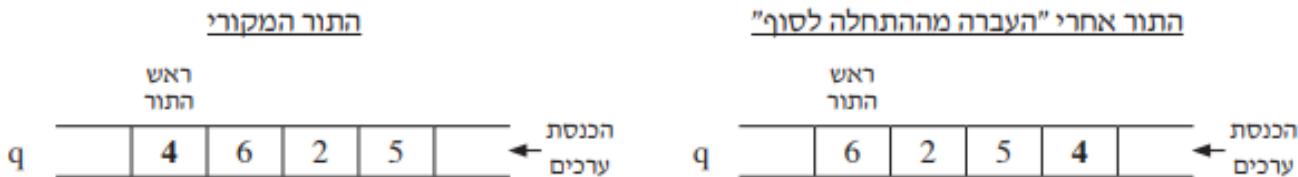
- המשך בשקופית  
 הבאה



# המשך - שאלת מבגרות 2021

ב. "העברה מהתחלת לסוף" היא העברת מספר מראש התור לסופו.

דוגמאות:

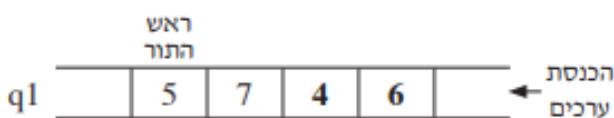


כתב פוליה חיונית שנקראה `isSimilar` בשפת C# או `Java` אם המקלט שני תורים מטיפוסם, `q1` ו`q2`. הפעולה מחזירה `true` אם התורים `q1` ו`q2` זהים – בין שהם זהים כמו שהם ובין שהם יהיו זהים לאחר שנבצע ב-`q1` "העברה מהתחלת לסוף", בעמיה או יותר. אחרת הפעולה מחזירה `false`.

דוגמאות: הפעולה תחזיר `true` בעבור שני התורים `q1` ו`q2` שלפנינו:



את מכיוון שלאחר שבתור `q1` נבצע פעמיים "העברה מהתחלת לסוף", הוא יהיה זהה לתור `q2`, ויראה כך:



הערות: – חוובה להשתמש בפעולה שכתבת בסעיף א.

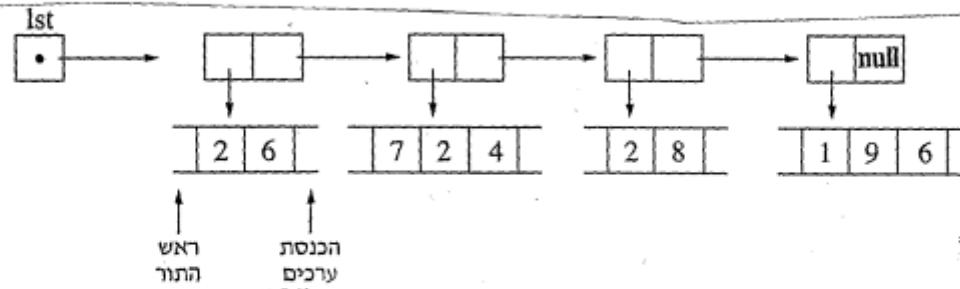
– אין צורך לשמנוע על מבנה התורים המקורי שהתקבל.



# שאלה מבגרות 2019

ב. נתונה שרשרת חוליות ובה כל חוליה מכילה "תור מספר".

לדוגמא: שרשרת החוליות שלפניך מייצגת את המספרים 691 , 196 , 82 , 427 , 62 .



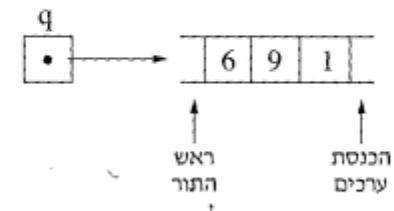
כתב פעלת בשם `bigNumber` ב- C# או `bigNumber` Java או ב- C#, המתקבלת הפניה `lst` לשרשרת החוליות,

ומוחזירה את המספר הגדל ביותר המוצג בשרשרת החוליות.

עבור שרשרת החוליות שתוארה בדוגמה הפעולה מוחזירה את המספר 691 .

חובה להשתמש בפעולה שהוגדרה בסעיף א.

5. א. "תור מספר" הוא תור של ספרות בין 1 ל- 9 (כולל), המציג מספר שלם – האיבר הראשון (ראש התור) הוא ספרת היחידות, האיבר השני הוא סhort העשורות וכן הלאה.  
הנה שמספר הספרות האפשרי בתור לא גדול ממספר הספרות שיכל להכיל משתנה מטיפוס `int` .  
לדוגמא: התור שלפניך מייצג את המספר 196 .



כתב פעלת בשם `toNumber` Java או ב- C# ב- C#, מקבלת "תור מספר" – `q` , ומוחזירה את המספר המיוצג בתור.  
הערה: אין חובה לשמור על מבנה התור.

# שאלה מבגרות 2006

לפניך פעולה:

תור _לפי_ שכיחות (Q)
הפעולה מקבלת תור Q המכיל מספרים שלמים, ומחזירה תור חדש. בעבור כל מספר בתור Q, יהיה בתור החדש שני איברים: האיבר הראשון מכיל את המספר מהתור Q, והאיבר השני מכיל את מספר הפעמים שהוא מופיע בתור Q. בעבור מספר המופיע יותר מפעם אחת בתור Q, יהיה זוג אחד בלבד בתור החדש. הנחה: התור Q מאותחל. הערה: אין לשנות את התור Q.

לדוגמא:

נתון התור Q (משמאל לימין):

1	4	4	1	5	-9	1	-9	-9
---	---	---	---	---	----	---	----	----

התור שיוחזר לאחר זימון הפעולה **תור \_לפי\_ שכיחות (Q)** יהיה (משמאל לימין):

1	3	4	2	5	1	-9	3
---	---	---	---	---	---	----	---

א. כתוב אלגוריתם, שיממש את הפעולה **תור \_לפי\_ שכיחות (Q)**.

אפשר להשתמש בפעולות המשק תור ובפעולה **העתק\_טור (Q)** שלפניך, בלי למכש אותן.

העתק_טור (Q)
הפעולה מקבלת תור Q וממחזירה תור חדש זהה לו. הנחה: Q מאותחל. סיבוכיות זמן הריצה: (n)O, כ הוא מספר האיברים בתור Q.

ב. מהי סיבוכיות זמן הריצה של האלגוריתם שכתבת בסעיף א? נמק את תשובתך.

הנה סיבוכיות זמן הריצה של כל אחת מפעולות המשק תור היא (1)O .

/המשק בעמוד 3/

+

הפעולה מקבלת תור לא ריק, המכיל מספרים שלמים.  
/\* הפעולהמחזירה...\*/

```
public static int Sod1(Queue<int> q)
{
    int i = q.Remove();
    int result = i;

    if (!q.IsEmpty())
    {
        int j = Sod1(q);
        if (result > j)
            result = j;
    }
    q.Insert(i);
    return result;
}
```

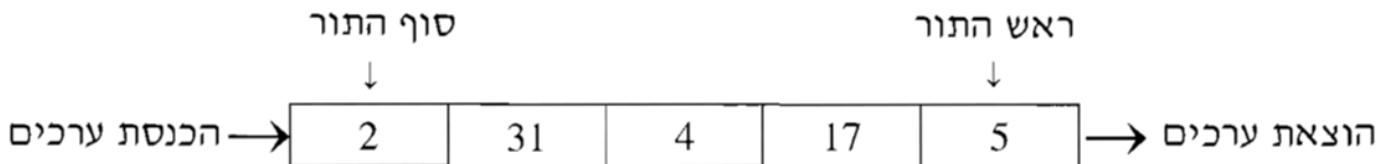
הפעולה מקבלת מספר שלם גדול מ- 0 או שווה לו  
/\* הפעולהמחזירה...\*/

```
public static int Sod2 (int i)
{
    if (i == 0)
        return 0;
    int a = i % 10;
    int b = Sod2(i / 10);
    if (a > b)
        return a;
    return b;
}
```

# שאלה מבגרות 2009

# שאלות מבחן 2009 - המשך

נתון התור Queue<int> myQueue מטיפוס



- ב.** מה מבצעת הפעולה `Sod1( queue )`? עברו תור `queue` לא ריק  
**ג.** רשום את התור המתקין בתום המעקב. ציין את ראש התור ואת סוף התור  
**א.** מה יחזיר הזימון `(myQueue Sod1)`? רשום את המעקב.

מטען Queue<int>

- ג. מה יחזיר היזיון `Sod2(17852)` ? רשום את המקבב.

ד. מה מבצעת הפולה `Sod2(k)` בעבר מסטר `k` גדול מ- 0 מטיפוס שלם

ה. מה מבצעת הפולה `Sod2(Sod1(queue))` בעבר תור `queue` לא ריק

матיפוס `<int>` המכיל רק מספרים שלמים וגדולים מ- 0 ?

4. איבר **מחליף** **כיוון** הוא מספר בסדרת מספרים שבו משתנה סדר המיוון בין איברים סמוכים מעליה לירידה או מירידה לעלייה. איבר **מחליף** **כיוון** שיקן גם לרצף של מספרים הממוינים בסדר עולה וגם לרצף של מספרים הממוינים בסדר יורד.

לדוגמה: בסדרה 8 1 2 5 7 4 2 המספרים 7 ו- 1 הם **איברים מחליפים כיוון**.

א. כתוב באחת מרבע השפות – פסקל, C , C# , Java – תת-תיכנית/פעולה חיצונית (סטטיטית), שתתקבל מחסנית לא ריקה של מספרים שלמים, שאין בה מספרים זהים צמודים זה לזה. התת-תיכנית/פעולה תכניס למחסנית אחורי כל **איבר מחליף כיוון** איבר זהה לו.

דוגמאות:

<table border="1" style="border-collapse: collapse; width: 100%; height: 100%;"> <tr><td style="width: 10px;">2</td></tr> <tr><td>1</td></tr> <tr><td>2</td></tr> <tr><td>3</td></tr> <tr><td>3</td></tr> <tr><td>2</td></tr> <tr><td>1</td></tr> </table>	2	1	2	3	3	2	1	<table border="1" style="border-collapse: collapse; width: 100%; height: 100%;"> <tr><td style="width: 10px;">2</td></tr> <tr><td>1</td></tr> <tr><td>1</td></tr> <tr><td>2</td></tr> <tr><td>3</td></tr> <tr><td>3</td></tr> <tr><td>2</td></tr> <tr><td>1</td></tr> </table>	2	1	1	2	3	3	2	1	
2																	
1																	
2																	
3																	
3																	
2																	
1																	
2																	
1																	
1																	
2																	
3																	
3																	
2																	
1																	
<table border="1" style="border-collapse: collapse; width: 100%; height: 100%;"> <tr><td style="width: 10px;">3</td></tr> <tr><td>7</td></tr> <tr><td>7</td></tr> <tr><td>3</td></tr> <tr><td>3</td></tr> <tr><td>7</td></tr> <tr><td>7</td></tr> <tr><td>3</td></tr> </table>	3	7	7	3	3	7	7	3	<table border="1" style="border-collapse: collapse; width: 100%; height: 100%;"> <tr><td style="width: 10px;">3</td></tr> <tr><td>7</td></tr> <tr><td>7</td></tr> <tr><td>3</td></tr> <tr><td>3</td></tr> <tr><td>7</td></tr> <tr><td>7</td></tr> <tr><td>3</td></tr> </table>	3	7	7	3	3	7	7	3
3																	
7																	
7																	
3																	
3																	
7																	
7																	
3																	
3																	
7																	
7																	
3																	
3																	
7																	
7																	
3																	
<table border="1" style="border-collapse: collapse; width: 100%; height: 100%;"> <tr><td style="width: 10px;">9</td></tr> <tr><td>6</td></tr> <tr><td>5</td></tr> <tr><td>4</td></tr> </table>	9	6	5	4	<table border="1" style="border-collapse: collapse; width: 100%; height: 100%;"> <tr><td style="width: 10px;">9</td></tr> <tr><td>6</td></tr> <tr><td>5</td></tr> <tr><td>4</td></tr> </table>	9	6	5	4								
9																	
6																	
5																	
4																	
9																	
6																	
5																	
4																	

ב. מהי סיבוכיות זמן הריצה של התת-תיכנית/הפעולה שכתבת? נמק את תשובהך.

הערה:

# בגרות 2008 - מחסנית







