



תאריך: מרץ-2026

צבי מלמד – דף תירגול מדעי המחשב כיתה יא-1 תשפ"ו



**מבוא למדעי המחשב – כיתה יא-1**  
**שאלות לתרגול מבחינות הבגרות**  
**הנושא: רשימות**

**בחינה : 2016 מועד ב'**

**שאלה מספר: 4**

**השאלה עוסקת ב-:**

- **רשימה שכל השלישים שבה זהים**

**פרק שני (50 נקודות)**

**שים לב:** בכל שאלה שנדרש בה מימוש אתה יכול להשתמש בפעולות של המחלקות תור, מחסנית, עץ בינרי וחוליה, בלי לממש אותן. אם אתה משתמש בפעולות נוספות, עליך לממש אותן.

ענה על שתיים מהשאלות 4-6 (לכל שאלה – 25 נקודות).

4. רשימה L תיקרא **משולשת** אם היא מקיימת את התנאים האלה:

\* הרשימה אינה ריקה.

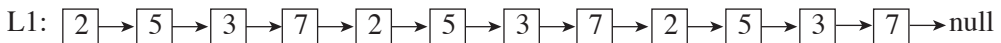
\* מספר האיברים בה מתחלק ב-3 בלי שארית.

\* האיברים בשליש הראשון של הרשימה מכילים את אותם ערכים שמכילים האיברים

בשליש השני של הרשימה ואותם ערכים שמכילים האיברים בשליש השלישי של הרשימה.

הערכים מסודרים באותו סדר בכל אחד מהשלישים.

לדוגמה: הרשימה L1 שלפניך היא רשימה **משולשת** באורך 12.



כתוב ב- C# או ב- Java פעולה חיצונית שתקבל רשימה L שהאיברים שלה הם מטיפוס שלם.

אם L היא רשימה **משולשת**, הפעולה תחזיר true .

אחרת – הפעולה תחזיר false.

/המשך בעמוד 8/

תאריך: מרץ-2026

**מבוא למדעי המחשב – כיתה יא-1**  
**שאלות לתרגול מבחינות הבגרות**  
**הנושא: רשימות**

בחינה : 2018

שאלה מספר: 5

**השאלה עוסקת ב-:**

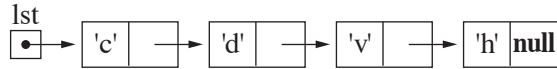
- שאלת מעקב רקורסיה
- רשימה של תווים עם 'a', 'b' סמוכים וקריאה לפעולה רקורסיבית

C# לפותרים בשפת

נתונה הפעולה Sod1 :

```
public static Node <char> Sod1(Node <char> lst, char ch)
{
    if (lst == null)
        return null;
    if (lst.GetValue() == ch)
        return lst;
    return Sod1(lst.GetNext(), ch);
}
```

א. (1) עקוב אחר הפעולה וכתוב מה יוחזר עבור  $ch = 'v'$  וההפניה lst לשרשרת חוליות של תווים:



(2) מהי מטרת הפעולה Sod1 ?

(3) מהי סיבוכיות זמן הריצה של הפעולה Sod1 ? נמק.

ב. נתונה הפעולה Sod2 :

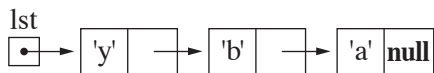
```
public static bool Sod2(Node <char> lst)
{
    if (Sod1(lst,'a') != null && Sod1(lst,'b') != null)
        return true;
    return false;
}
```

מהי מטרת הפעולה Sod2 ?

ג. כתוב פעולה בוליאנית המקבלת הפניה לשרשרת חוליות של תווים ומחזירה true אם מופיעות בה שתי חוליות

סמוכות שערכיהן 'b' 'a' או 'a' 'b'. אחרת – הפעולה מחזירה false.

דוגמה לשרשרת חוליות שמופיעים בה 'b' 'a' ברצף: דוגמה לשרשרת חוליות שמופיעים בה 'b' 'a' ברצף:



דוגמה לשרשרת חוליות ש- 'b' 'a' ו- 'a' 'b' אינם מופיעים בה ברצף:



עליך להשתמש בפעולה Sod1.

הנח שכל התווים שונים זה מזה.

תאריך: מרץ-2026

**מבוא למדעי המחשב – כיתה יא-1**  
**שאלות לתרגול מבחינות הבגרות**  
**הנושא: רשימות**

**בחינה : 2019**

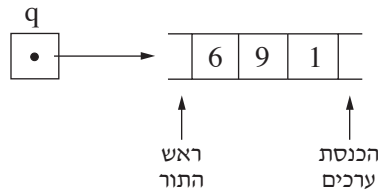
**שאלה מספר: 5**

**השאלה עוסקת ב-:**

- מעקב אחר רקורסיה
- פונקציה רקורסיבית ברשימות

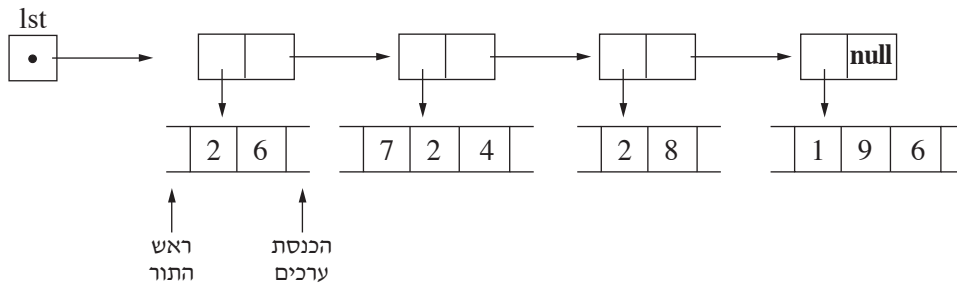
5. א. "תור מספר" הוא תור של ספרות בין 1 ל-9 (כולל), המייצג מספר שלם – האיבר הראשון (ראש התור) הוא ספרת האחדות, האיבר השני הוא ספרת העשרות וכן הלאה.

הנח שמספר הספרות האפשרי בתור לא גדול ממספר הספרות שיכול להכיל משתנה מטיפוס int. לדוגמה: התור שלפניך מייצג את המספר 196.



כתוב פעולה בשם toNumber ב-Java או ToNumber ב-C#, המקבלת "תור מספר" – q, ומחזירה את המספר המיוצג בתור. הערה: אין חובה לשמור על מבנה התור.

ב. נתונה שרשרת חוליות ובה כל חוליה מכילה "תור מספר". לדוגמה: שרשרת החוליות שלפניך מייצגת את המספרים 62, 427, 82, 691.



כתוב פעולה בשם bigNumber ב-Java או BigInteger ב-C#, המקבלת הפניה lst לשרשרת החוליות, ומחזירה את המספר הגדול ביותר המיוצג בשרשרת החוליות. עבור שרשרת החוליות שתוארה בדוגמה הפעולה מחזירה את המספר 691. חובה להשתמש בפעולה שהוגדרה בסעיף א.

תאריך: מרץ-2026

**מבוא למדעי המחשב – כיתה יא-1**  
**שאלות לתרגול מבחינות הבגרות**  
**הנושא: רשימות**

**בחינה : 2020 מועד ב'**

**שאלה מספר: 4**

**השאלה עוסקת ב-:**

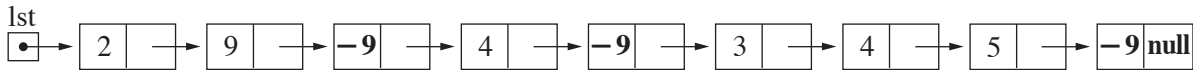
- **שרשרת מספרים חיוביים**
- **יצירת מספר מרשימה של ספרות**

**פרק שני** (40 נקודות)

**שים לב:** בכל שאלה שנדרש בה מימוש, תוכל להשתמש בפעולות של המחלקות: תור, מחסנית, עץ בינרי וחוליה, בלי לממש אותן. אם תשתמש בפעולות נוספות, יהיה עליך לממש אותן.

ענה על שתיים מן השאלות 4-6. (לכל שאלה – 20 נקודות)

- 4. • **"שרשרת מספרים שלמים חיוביים"** היא שרשרת חוליות שכל חוליה בה מכילה מספר שלם הגדול מ-0.
  - **"שרשרת ספרות"** היא שרשרת חוליות שכל חוליה בה מכילה ספרה בין 0 ל-9 (כולל) או את המספר (-9). כל רצף ספרות בשרשרת מייצג מספר: הספרה הראשונה מייצגת את האחדות, הספרה השנייה את העשרות וכן הלאה. לאחר כל רצף של ספרות מופיע המספר (-9), והוא מסמן סוף של מספר בשרשרת.
- לפניך דוגמה ל"שרשרת ספרות" המייצגת את המספרים: 92, 4, 543.



כתוב פעולה חיצונית בשפת Java בשם buildDigit או בשפת C# בשם BuildDigit, המקבלת הפניה lst שאינה null ל"שרשרת מספרים שלמים חיוביים". הפעולה תחזיר "שרשרת ספרות" המייצגת את המספרים שב"שרשרת מספרים שלמים חיוביים" לפי הסדר.

תאריך: מרץ-2026

**מבוא למדעי המחשב – כיתה יא-1**  
**שאלות לתרגול מבחינות הבגרות**  
**הנושא: רשימות**

**בחינה : 2024**

(טופס: 899271)

**שאלה מספר: 4**

**השאלה עוסקת ב-:**

- עבודה עם מחלקות (BusStation), כולל גישה למערך בתוך המחלקה.
- נתון מערך של אובייקטים, ליצור רשימה מקושרת ללא חזרות על בסיס הנתונים במערך.

4. נתונה המחלקה **BusStation** – תחנת אוטובוס, ולה שלוש תכונות:

- num – מספר התחנה, מטיפוס שלם
- arr – מערך מטיפוס שלם בגודל 10, המכיל את מספרי קווי האוטובוס שעוצרים בתחנה. בתחנה עוצרים עד 10 קווים, והם נשמרים ברצף מתחילת המערך.
- amount – כמות קווי האוטובוס שעוצרים בתחנה בפועל, מטיפוס שלם. בתחנה עוצר קו אוטובוס אחד לפחות. הניחו שלתכונות המחלקה יש פעולות get/Get ו set/Set.

א. ממשו את הפעולה שלפניכם השייכת לממשק המחלקה **BusStation**:

**Java** – public boolean isStopping (int n)

**C#** – public bool IsStopping (int n)

הפעולה מקבלת מספר קו אוטובוס – n מטיפוס שלם. הפעולה מחזירה true אם קו האוטובוס עוצר בתחנה, ואחרת מחזירה false.

ב. נתון מערך arr מטיפוס **BusStation**, ובו כל תחנות האוטובוס בעיר מסוימת. ידוע שכמה מקווי האוטובוס עוצרים

בכל אחת מן התחנות בעיר, ושאר הקווים עוצרים רק בחלק מן התחנות.

כתבו פעולה חיצונית ששמה **allStations** בשפת **Java** או **AllStations** בשפת **C#**, המקבלת את המערך **arr**.

הפעולה מחזירה שרשרת חוליות מטיפוס שלם שבכל חוליה מופיע אחד ממספרי הקווים שעוצרים בכל התחנות בעיר (כל קו כזה יופיע פעם אחת בלבד בשרשרת).

הערה: אין חשיבות לסדר הקווים בשרשרת.

תאריך: מרץ-2026

**מבוא למדעי המחשב – כיתה יא-1**  
**שאלות לתרגול מבחינות הבגרות**  
**הנושא: רשימות**

**בחינה : 2024**

(טופס: 899271)

**שאלה מספר: 5**

**השאלה עוסקת ב-:**

- רשימה של מספרים שלמים ללא חזרות, והאם קיימת "תת-סדרה נגדית" עבור מספר מסוים.
- מציאת תת-סדרה נגדית המקסימלית.

5. "תת־סדרה נגדית" היא רצף של מספרים המתחיל במספר כלשהו ומסתיים במספר הנגדי והשווה לו בערך המוחלט שלהם (כלומר מתחיל במספר חיובי ומסתיים במספר השלילי המקביל לו או מתחיל במספר שלילי ומסתיים במספר החיובי המקביל לו). הרצף נמצא בתוך סדרה של מספרים.

לדוגמה: בסדרת המספרים:  $17, 3, 1, -4, 5, 6, 4, -3, 7, 23, -99$  יש שתי תת־סדרות נגדיות:

i.  $3, 1, -4, 5, 6, 4, -3$

ii.  $-4, 5, 6, 4$

נתונה שרשרת חוליות lst מטיפוס שלם, המכילה מספרים חיוביים ושליליים שאינם 0, וכולם שונים זה מזה (כלומר אין שני מספרים זהים בשרשרת).

א. כתבו פעולה ששמה width בשפת Java או Width בשפת C# המקבלת את השרשרת lst ומספר שלם num (חיובי או שלילי) המופיע בשרשרת.

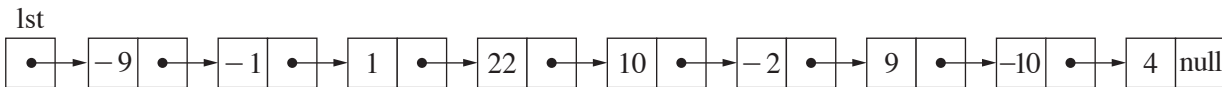
הפעולה תחזיר את אורך ה"תת־סדרה נגדית" שהמספר num מתחיל או מסיים (האורך כולל את המספרים בקצוות).

אם המספר הנגדי ל־num אינו מופיע בשרשרת, הפעולה תחזיר 1 - .

הערות: - חובה לשמור על השרשרת lst.

- אין להשתמש בסעיף זה במעריך. פתרון הכולל שימוש במעריך לא יזוכה בנקודות.

דוגמאות: עבור השרשרת lst שלפניכם:



עבור  $num = 9$  הפעולה תחזיר 7 (התת־סדרה:  $9, -2, 10, 22, 1, -1, -9$ ).

עבור  $num = -1$  הפעולה תחזיר 2 (התת־סדרה:  $1, -1$ ).

עבור  $num = 22$  הפעולה תחזיר 1 - (משום שהמספר הנגדי לו,  $-22$ , אינו מופיע בשרשרת).

ב. כתבו פעולה ששמה longest בשפת Java או Longest בשפת C# המקבלת את השרשרת lst. הפעולה תחזיר את

אורך ה"תת־סדרה נגדית" הגדולה ביותר. אם אין בשרשרת שום "תת־סדרה נגדית", הפעולה תחזיר 1 - .

אפשר להשתמש בפעולה שכתבתם בסעיף א.

הערות: - בפעולה זו אין חובה לשמור על השרשרת lst.

- אין להשתמש בסעיף זה במעריך. פתרון הכולל שימוש במעריך לא יזוכה בנקודות.

לדוגמה: עבור השרשרת lst שבדוגמה לעיל הפעולה תחזיר 7.

הסבר: ה"תת־סדרה נגדית" המתחילה במספר 9 - ומסתיימת במספר 9 מכילה שבעה מספרים והיא הגדולה ביותר.