

בהצלחה ל: _____ כתה: _____ שם המורה: _____

מתכונת מדעי המחשב תשפ"ד - שאלון 899381

הוראות לנבחן/ת

1. משך הבחינה : שלוש שעות ו-27 דקות (3 שעות ו 52 דקות עם תוספת זמן)

2. מבנה השאלון ומפתח הערכה :

בשאלון זו שלושה פרקים :

פרק ראשון - בפרק זה שלוש שאלות, יש לענות על שתיים.

שאלה 1 שאלת חובה – 10 נק'

שאלה 2 או שאלה 3 – 15 נק'

פרק שני ושלישי - בפרקים אלו שש שאלות, יש לענות על שלוש.

לכל שאלה 25 נקודות ($3 * 25$) – 75 נק'.

סה"כ 100 נק'.

3. חומר עזר מותר בשימוש : כל חומר עזר, חוץ ממחשב שניתן לתכנות וטלפון נייד, אין להעביר ספרים ודפים בין הנבחנים.

4. יש לציין את מספרי השאלות שעליהן בחרתם לענות :

5.

פרק ראשון : שאלות _____1, _____, _____,

פרק שני : שאלות _____, _____, _____, _____,

פרק שלישי : שאלה _____, _____

ב ה צ ל ח ה !

פרק ראשון - יסודות

שאלה 1 – חובה – 10 נק'

נתונה המחלקה "מילה במשפט" **WordInSen** שיש לה 2 תכונות:

- מילה, String word
- מיקומה של המילה במשפט – מספר שלם בין 1 לאורך המשפט – int place

מערך "משפט תקין" הוא מערך של עצמים מטיפוס **WordInSen**, שיכול להוות משפט תקין, כלומר שכל המיקומים של המילים מופיעים פעם אחת בלבד, ושהם בסדר עוקב מ-1 ועד אורך המשפט.

לדוגמא: המערך הבא הוא מערך "משפט תקין"

0	1	2	3	4	5
["the", 3]	["ever", 6]	["We", 1]	["are", 2]	["students", 5]	["best", 4]

והמערך הבא הוא לא מערך "משפט תקין", מכיוון שמיקום 4 לא מופיע, ואילו מיקום 3 מופיע פעמיים.

0	1	2	3	4	5
["the", 3]	["ever", 6]	["We", 1]	["are", 2]	["students", 5]	["best", 3]

סעיף א: כתבו פעולה חיצונית בשם **isCorrect** המקבלת מערך של עצמים מטיפוס **WordInSen** ומחזירה true אם מדובר ב"מערך משפט תקין". אחרת יוחזר false.

סעיף ב: כתבו פעולה חיצונית בשם **fullSentence** המקבלת מערך של עצמים מטיפוס **WordInSen**. הפעולה תחזיר מחרוזת, המהווה את המשפט שמסתתר במערך. אם המערך אינו תקין – תוחזר מחרוזת ריקה.

יש לענות על אחת מהשאלות 2-3

שאלה 2 - 15 נק'

המחלקה מחליקה על הקרח **IceSkater** כוללת 3 תכונות:

- שם - name - מחרוזת
- גיל - age - מספר שלם
- סגנון החלקה - style - מחרוזת

הניחו קיימות פעולות **get, set** לכל התכונות במחלקה.

נבחרת מחליקות, המחלקה **SkaterTeam** יכולה לכלול עד 20 מחליקות לכל היותר, והיא כוללת את התכונות הבאות:

- שם הנבחרת – `String teamName`
- מערך חד מימדי מטיפוס **IceSkater** בשם `arr`.

הניחו קיימות פעולות **get, set** לכל התכונות במחלקה.

סעיף א: כתבו את כותרת המחלקה **SkaterTeam**, התכונות והפעולה הבונה.

סעיף ב: מועמדת לנבחרת יכולה להתקבל רק אם סגנון ההחלקה שלה, קיים בנבחרת רק אצל פחות משליש מהמחליקות הקיימות.

כתבו פעולה **פנימית** במחלקה **SkaterTeam**, בשם **addSkater**, שתקבל עצם מטיפוס **IceSkater**, ותוסיף את המחליקה למערך, אם יש מקום במערך ואם היא עומדת בתנאי הקבלה.

סעיף ג: כתבו פעולה **חיצונית** בשם **youngest**, שמקבלת מערך עצמים מטיפוס **IceSkater** וסגנון החלקה. הפעולה תחזיר את שם המחליקה הצעירה ביותר עם אותו סגנון החלקה שהתקבל כפרמטר. אם סגנון ההחלקה לא קיים, תוחזר מחרוזת ריקה.

שאלה 3 - 15 נק'

נתונה מחלקת גיבור/ת-על **Superhero** המכילה שלוש תכונות:

- abilities - כמות סוגי יכולות על – מספר שלם.
- name – שם - מטיפוס מחרוזת המתארת את שם הגיבור/ה
- savings – הצלות - מטיפוס שלם – כמות האנשים שהגיבור/ה הצילה.

הניחו קיימות פעולות **get, set** לכל התכונות במחלקה.

-

סעיף א: כתבו את כותרת המחלקה ואת תכונותיה.

סעיף ב: כתבו פעולה בונה מעתיקה במחלקה **Superhero**, המקבלת שם של גיבור/ת-על ועוד עצם מטיפוס **Superhero**, ויוצרת גיבור/ת-על דומה, רק שהשם הוא המחרוזת החדשה שהתקבלה כפרמטר, ושאר ערכי התכונות זהים לאלה של העצם שהתקבל.

-

סעיף ג: נתונה מחלקת "ליגת הצדק" **JusticeLeague**, המכילה תכונה אחת: מערך חד מימדי של גיבור/יות-על - **heroes**.

כדי לצרף גיבור/ת-על חדשה לליגת הצדק, יש לוודא שיש מקום במערך, ושלכל גיבור/ות-העל יהיו 2 יכולות-על לכל היותר.

הניחו קיימות פעולות **get, set** לכל התכונות במחלקה.

כתבו פעולה פנימית במחלקה **JusticeLeague** בשם **canJoin**, שתקבל עצם מטיפוס **Superhero**, ותוסיף אותו למערך אם ניתן לצרף אותו לליגת הצדק. הפעולה תחזיר **true** אם העצם התוסף למערך. אחרת יוחזר **false**.

סעיף ד: כתבו פעולה חיצונית **bestSaver**, המקבלת עצם מטיפוס **JusticeLeague** ומחזירה עצם מטיפוס **Superhero** עם כמות ההצלות הגבוהה ביותר. הניחו שיש רק אחד כזה.

יש לענות על 3 שאלות מתוך שאלות 4-9

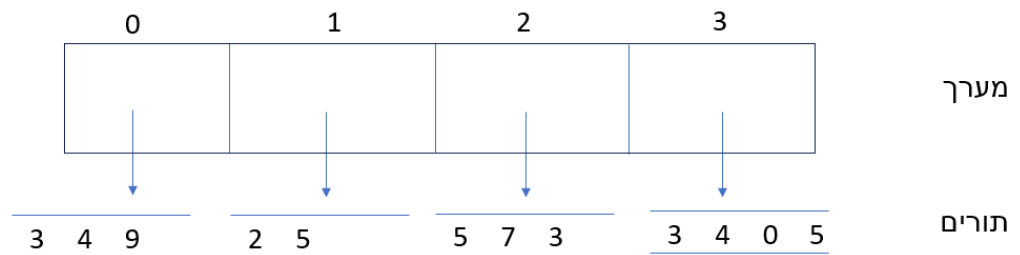
פרק שני – מבני נתונים

שאלה 4:

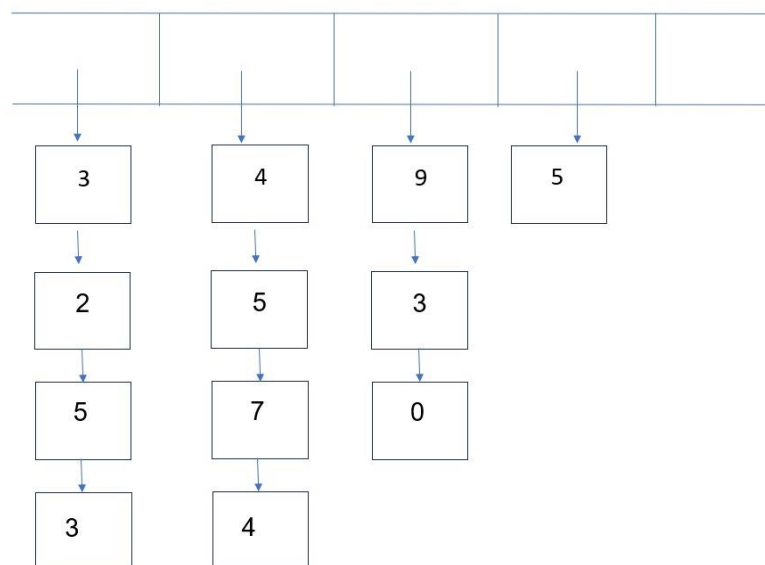
סעיף א: יש לכתוב פעולה חיצונית בשם **wide**, המקבלת מערך לא ריק של תורים לא ריקים מטיפוס מספר שלם, ומחזירה תור של שרשרת חוליות מטיפוס מספר שלם, לפי הכללים הבאים:

- שרשרת חוליות ראשונה בתור תכלול את כל הערכים של האיבר הראשון בכל תור.
- שרשרת חוליות שנייה בתור תכלול את כל הערכים של האיבר השני בכל תור, אם קיים איבר כזה.
- שרשרת חוליות שלישית בתור תכלול את כל הערכים של האיבר השלישי בכל תור, אם קיים איבר כזה, וכן הלאה לשרשרת הרביעית, חמישית...
- אם בעת בניית השרשרת אין איבר מתאים בתור (התור קצר יותר), לא תכלול השרשרת כל איבר השייך לתור זה.

לדוגמא, בקבלת מערך התורים הבא:



יתקבל תור הרשימות להלן:



סעיף ב: מהי סיבוכיות הפעולה שכתבת בסעיף א? חובה לנמק

שאלה 5:

רשימה "מדופלמת" הינה רשימה מטיפוס מספר שלם, אשר כל מספר ברשימה גדול משלושה מספרים או יותר מהמספרים שאחריו ברשימה (לא חייב מספרים ברצף), אם לא קיימים 3 מספרים אחרי מספר מסוים ברשימה, אז מספר זה יהיה גדול מכל המספרים הבאים אחריו (רשימה ריקה או רשימה עם מספר אחד הינה "רשימה מדופלמת").

דוגמא לרשימה לא מדופלמת:	דוגמא לרשימה מדופלמת:
<ul style="list-style-type: none"> • 122 גדול מ-14,29,30,6,25,7,2,1 • 14 גדול מ-6,7,2,1 • 29 גדול מ-6,25,7,2,1 • 30 גדול מ-6,25,7,2,1 • 6 גדול רק מ-2,1 	<ul style="list-style-type: none"> • 122 גדול מ-14,29,30,6,25,7,2,1 • 14 גדול מ-6,7,2,1 • 29 גדול מ-6,25,7,2,1 • 30 גדול מ-6,25,7,2,1 • 6 גדול מ-3,2,1 • 25 גדול מ-3,2,1 • 3 גדול מ-2,1 • 2 גדול מ-1

ניתן להשתמש בפעולה `sizeList` המקבלת רשימה ומחזירה את מספר החוליות ברשימה מבלי לממש אותה.

סעיף א: יש לכתוב פעולה בשם `biggerNumbers`, המקבלת רשימה ומספר ומחזירה כמה מספרים ברשימה גדולים ממספר זה.

סעיף ב: כתבו פעולה `checkList`, המקבלת רשימה של מספרים שלמים. הפעולה תחזיר `true` אם הרשימה הינה רשימה מדופלמת, אחרת `false`. חובה להשתמש בפעולה מסעיף א.

סעיף ג: מהי סיבוכיות הפעולה שכתבת בסעיף ב ? חובה לנמק.

שאלה 6:

להלן הפעולה **what2** המזמנת את הפעולה **what1**

הפעולה what1

טענת כניסה: הפעולה מקבלת תור q של מספרים שלמים, מספר טבעי n המציין את מספר איברי התור ומספר שלם a .

```
public static int what1 (Queue<Integer> q, int n, int a)
{
    while (n > 0)
    {
        int b = q.remove();
        n--;
        if (b < a)
        {
            q.insert(a);
            a = b;
        }
        else
            q.insert(b);
    }
    return a;
}
```

הפעולה what2

טענת כניסה: הפעולה מקבלת תור q של מספרים שלמים ומספר טבעי n המציין את מספר איברי התור. x, y הם משתנים מטיפוס שלם.
הנחה: התור q אינו ריק

```
public static void what2 (Queue<Integer> q, int n)
{
    if (n != 1)
    {
        int x = q.remove();
        int y = what1 (q, n-1, x);
        what2 (q, n-1);
        q.insert(y);
    }
}
```

סעיף א: יש לבצע ולהציג מעקב אחר ביצוע הפעולה **what1**, עבור התור q והנתונים הבאים:

$$a = 2, n = 4, q = \{7, 9, 1, 4\}$$

סעיף ב: מה מבצעת הפעולה **what1**? השלימו את טענת היציאה.

המשך השאלה בעמ' הבא ←

סעיף ג: יש לרשום דוגמא לזימון הפעולה **what1**, כך שהתור q לא ישתנה. נמקו בקצרה את תשובתכם/ן.

סעיף ד: יש לבצע ולהציג מעקב אחר ביצוע הפעולה **what2**, עבור התור q והנתונים הבאים:

$$n = 4, q = \{6, 8, 11, 3\}$$

סעיף ה: מה מבצעת הפעולה **what2**? השלימו את טענת היציאה.

שאלה 7:

"תור שווה סכומים", הוא תור של מספרים שלמים, המכיל מספר אי זוגי של איברים, כך שכל שני איברים קיצוניים (ראשון+אחרון, שני+לפני האחרון וכו') שווים בסכומם לאיבר האמצעי של התור. לדוגמא: התור הבא הוא "תור שווה סכומים" -

האיבר האמצעי הוא 25 וכל זוג איברים קיצוניים שווה ל 25

18	3	10	13	4	25	21	12	15	22	7
----	---	----	----	---	----	----	----	----	----	---

$$18+7 = 25$$

$$3+22 = 25$$

$$10+15 = 25$$

$$13+12 = 25$$

$$4+21 = 25$$

הניחו שקיימת פעולה **removeLast**, המקבלת תור לא ריק של מספרים שלמים. הפעולה מוציאה ומחזירה איבר מסוף התור. הפעולה שומרת על התור המקורי, למעט האיבר שהוחזר. סיבוכיות הפעולה היא $O(n)$. **אין צורך לממש פעולה זו.**

סעיף א: כתבו פעולה **equalSums**, המקבלת תור של מספרים שלמים ומחזירה true אם התור הוא "תור שווה סכומים", אחרת יוחזר false.

סעיף ב: כתבו פעולה **removeEqualSums**, המקבלת רשימה של תורים של מספרים שלמים. הפעולה תמחק מהרשימה את כל התורים שהם אינם "שווה סכומים". **חובה להשתמש בפעולה מסעיף א'.**

סעיף ג: מהי סיבוכיות הפעולה שכתבת בסעיף ב' ? **חובה לנמק.**

פרק שלישי – מודלים חישוביים

שאלה 8:

בשאלה זו שני סעיפים. אין קשר ביניהם. יש לענות על שניהם.

סעיף א:

להלן השפות L_1 - L_4 מעל הא"ב $\{a,b,c\}$:

$$L_1 = \{ c^i b^k a^k \mid i, k \geq 0 \}$$

$$L_2 = \{ w \cdot R(w) \mid w \in \{a,b,c\}^* \}$$

$$L_3 = \{ a^i b^i c^k \mid i, k \geq 0 \}$$

$$L_4 = \{ a^i b^i c^k \mid i > 0, k > 0 \}$$

א. בעבור כל אחת מהשפות שבתת סעיפים א-2:

א1. $L_1 \cap L_3$

א2. $L_2 \cap L_3$

רשמו מילה לא ריקה השייכת לשפה, שהאורך שלה הוא הקצר ביותר. תארו כל אחת מהשפות בצורה הקצרה ביותר.

ב. מהי השפה $(L_1 \cdot L_3) \cap L_2$? נמקו.

ג. לכל אחד מהתת-סעיפים ג1-ג2, קבעו אם הוא נכון או לא נכון. נמקו את קביעתכם.

ג1. $R(L_2) = L_2$

ג2. $R(L_1) = L_3$

ד. מהי השפה $L_2 \cdot (L_1 \cap L_4)$? האם השפה רגולרית? נמקו.

סעיף ב:

להלן השפה L מעל הא"ב $\{a, b, c\}$:

$$L = \{ a^n b^m c^k b^k \mid n > m > 0, k > 0 \}$$

בנו אוטומט מחסנית שיקבל את שפה L .

שאלה 9:

בשאלה זו שני סעיפים. אין קשר ביניהם. יש לענות על שניהם.

סעיף א:

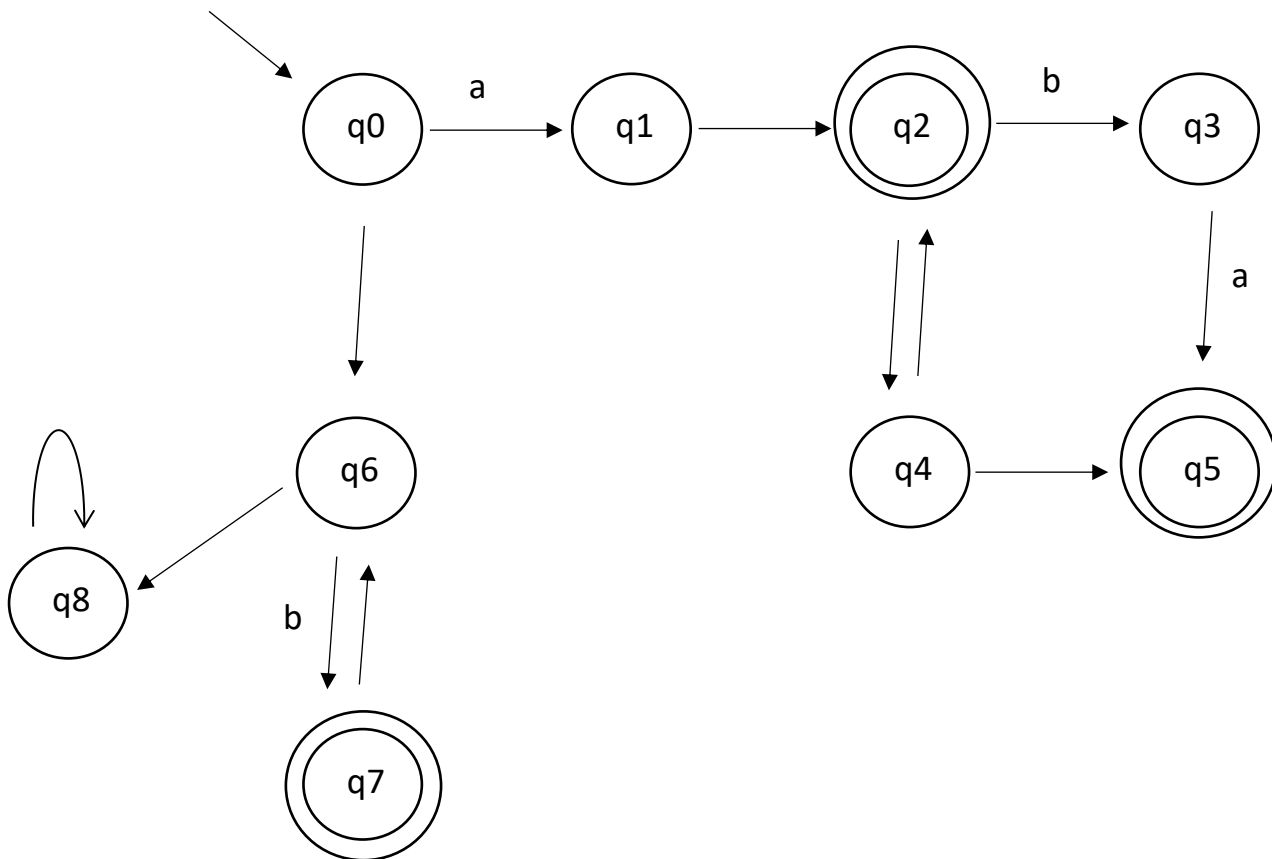
להלן השפה L מעל א"ב $\{a,b\}$ המוגדרת בצורה הבאה:

עבור כל המילים המתחילות ב- a , יהיה אורך המילה זוגי (גדול מאפס)

עבור כל המילים המתחילות ב- b , נקבל מילים מהצורה b^n - כאשר n זוגי וגדול מאפס.

להלן שרטוט חלקי של אוטומט סופי דטרמיניסטי מלא, המקבל את השפה L . בשרטוט חסרים מעברים

וסימני קלט. השרטוט מכיל את כל המצבים של האוטומט ואת המצבים המקבלים.



העתיקו את השרטוט למחברת והשלימו אותו, כך שהאוטומט יקבל את השפה L .

שימו לב! אין להוסיף לאוטומט או להוריד ממנו מצבים, ואין לסמן מצבים מקבלים נוספים.

סעיף ב:

בנו אוטומט מחסנית עבור השפה L מעל הא"ב $\{0,1\}$

$$L = \{ (0^n 1^k (20)^n 0)^p 1^p \mid n, p, k > 0, n \text{ זוגי-} k \text{ אי זוגי-} \}$$

p -מבטא את מספר החזרות על רצפים מהצורה $(0^n 1^k (20)^n 0)$