מדינת ישראל
משרד החינוך

סוג הבחינה: בגרות
מועד הבחינה: קיץ תשפ"ד, 2024
מספר השאלון: 899381

שימו לב: בבחינה זו יש הנחיות מיוחדות.
יש לענות על השאלות על פי הנחיות אלה.

מדעי המחשב

הוראות

א. משך הבחינה: שלוש שעות וחצי.

ב. מבנה השאלון ומפתח ההערכה: בשאלון זה שלושה פרקים.

פרק ראשון	–	(10×1) + (15×1)	–	25 נקודות
פרק שני פרק שלישי	–	(25×3)	–	75 נקודות
סך הכול	–		–	100 נקודות

שימו לב: אם תבחרו לענות על שאלות מן הפרק השלישי: בחרו בשאלות מתוך מסלול אחד בלבד.

ג. חומר עזר מותר בשימוש: כל חומר עזר, חוץ ממחשבון שיש בו אפשרות תכנות.

ד. הוראות מיוחדות:

(1) אם תבחרו לענות על שאלות **רק** מן הפרק הראשון והפרק השני, **רשמו על הכריכה החיצונית** של המחברת **ללא מסלול**, אחרת רשמו את **שם המסלול שלמדתם**.
המסלול הוא אחד משלושת המסלולים האלה: אלגוריתמים, מודלים חישוביים, תכנות מונחה עצמים.

(2) את כל התוכניות שיש לכתוב בשפת מחשב בפרקים הראשון והשני כתבו בשפה אחת בלבד – Java או C# .

הערה: לא יורדו נקודות אם תכתבו בתוכניות אות גדולה במקום אות קטנה או להפך.

יש לכתוב במחברת הבחינה בלבד. יש לרשום "טיוטה" בראש כל עמוד המשמש טיוטה.
כתיבת טיוטה בדפים שאינם במחברת הבחינה עלולה לגרום לפסילת הבחינה.

השאלות בשאלון זה מנוסחות בלשון רבים, אף על פי כן על כל תלמידה וכל תלמיד להשיב עליהן באופן אישי.

בהצלחה!

/המשך מעבר לדף/

השאלות

הערה: בכל שאלה שנדרשת בה קליטה, אין צורך לבדוק את תקינות הקלט.

לפותרים בשפת Java : בכל שאלה שנדרשת בה קליטה, הניחו שבתוכנית כתובה ההוראה:

```
Scanner input = new Scanner (System.in);
```

פרק ראשון (25 נקודות)

ענו על שאלה 1 – חובה (10 נקודות).

1. כתבו פעולה חיצונית ששמה biggestSum בשפת Java או BiggestSum בשפת C# , המקבלת מערך מטיפוס שלם – arr , שאין בו מספרים שליליים (המספרים גדולים או שווים ל־ 0). המספר 0 מופיע במערך פעמיים לפחות.

הפעולה תחזיר את סכום המספרים הגבוה ביותר המתקבל בין כל שני מספרי 0 במערך.

הניחו שיש מספרים חיוביים בין כל שני מספרי 0 במערך.

לדוגמה:

עבור המערך שלפניכם הפעולה תחזיר 17 .

	0	1	2	3	4	5	6	7	8	9	10	11
arr	33	0	**5**	**4**	0	**17**	0	**4**	**10**	0	5	14
			9			17		14				

הסבר: 17 הוא הסכום הגדול ביותר מבין סכומי המספרים המתקבלים בין כל שני מספרי 0 במערך (14, 17, 9).

/המשך בעמוד 3

ענו על <u>אחת</u> מן השאלות 2–3 (15 נקודות).

2. לאחר הבחירות, החל מזכיר הכנסת לשבץ את חברי הכנסת בוועדות הכנסת השונות.

לצורך כך הוגדרו המחלקות האלה: **Member** – חבר כנסת, ו־ **Committee** – ועדה.

למחלקה Member שתי תכונות:

- name – שם חבר הכנסת (הניחו שאין שני חברי כנסת שהשם שלהם זהה)
- isCoal – ערך בוליאני המקבל true אם חבר הכנסת שייך לקואליציה, ואחרת מקבל false .

למחלקה Committee שלוש תכונות:

- name – שם הוועדה
- members – מערך מטיפוס Member המכיל את חברי הכנסת המשובצים בוועדה
- count – כמות חברי הוועדה הנוכחיים בוועדה (חברי הוועדה נשמרים ברצף מתחילת המערך).

הניחו שקיימות פעולות get/Get ו־ set/Set לכל אחת מן התכונות של המחלקות.

א. כתבו את כותרות המחלקות Member ו־ Committee ואת תכונותיהן.

ב. כתבו פעולה פנימית במחלקה Committee ששמה total בשפת Java או Total בשפת C# , המקבלת פרמטר בוליאני – belong .

אם belong הוא true , הפעולה תחזיר את כמות חברי הכנסת בוועדה השייכים לקואליציה, אחרת (אם belong הוא false) הפעולה תחזיר את כמות חברי הכנסת בוועדה שאינם שייכים לקואליציה.

בשיבוץ חברי הכנסת לוועדות מזכיר הכנסת מקפיד על שני הכללים שלהלן:

- בכל ועדה כמות חברי הכנסת השייכים לקואליציה תהיה גדולה מכמות חברי הכנסת שאינם שייכים לקואליציה.
- בכל ועדה כמות חברי הכנסת תהיה קטנה מ־ 16 .

ג. ממשו את הפעולה החיצונית שלפניכם:

Java: public static int amount (Committee [] arr, Member m)

C#: public static int Amount (Committee [] arr, Member m)

הפעולה מקבלת מערך מלא – arr של כל ועדות הכנסת, וחבר כנסת מסוים – m , שעדיין לא שובץ בשום ועדה.

הפעולה תחזיר את כמות הוועדות שאפשר לשבץ בהן את חבר הכנסת (בהתאם לכללים של מזכיר הכנסת).

חובה להשתמש בפעולה שכתבתם בסעיף ב.

/המשך בעמוד 4

3. "מערך מכונות לייצור ברגים" הוא מערך מטיפוס שלם שבכל אחד מתאיו מופיע מספר **השניות** שנדרש למכונה אחת מסוימת כדי לייצר בורג **אחד** (ככל שמספר השניות קטן יותר, המכונה מייצרת ברגים במהירות גדולה יותר). המערך אינו ממוין.

לדוגמה: במערך שלפניכם למכונה באינדקס 0 נדרשת שנייה אחת בלבד כדי לייצר בורג אחד, למכונה באינדקס 1 נדרשות 9 שניות כדי לייצר בורג אחד ולמכונה באינדקס 2 נדרשות 3 שניות כדי לייצר בורג אחד.

	0	1	2
"מערך מכונות לייצור ברגים"	1	9	3

במפעל לייצור ברגים "הברגייה" יש כמה מכונות לייצור ברגים. המפעל שומר ב"מערך מכונות לייצור ברגים" – arr את מספר השניות הנדרש לכל אחת מן המכונות שלו לייצר בורג אחד. המפעל משתמש בכל המכונות לייצור ברגים באותו הזמן.

א. כתבו פעולה חיצונית ששמה total בשפת Java או Total בשפת C# , המקבלת את המערך arr , ומספר שלם של שניות – numSeconds .

הפעולה תחזיר את מספר הברגים סך הכול שהמפעל מייצר באמצעות כל המכונות שלו במשך numSeconds .

לדוגמה: עבור המערך שבדוגמה שלעיל ו־ numSeconds = 5 , הפעולה תחזיר 6 .

הסבר: תפוקת המכונות בפרק זמן של 5 שניות מפורטת לפניכם:

המכונה באינדקס 0 מייצרת חמישה ברגים (בכל שנייה בורג אחד),

המכונה באינדקס 1 אינה מייצרת אפילו בורג אחד (כי נדרשות לה 9 שניות כדי לייצר בורג אחד),

והמכונה באינדקס 2 מייצרת בורג אחד.

סך הכול: שישה ברגים (1 + 5).

ב. כתבו פעולה חיצונית ששמה minTime בשפת Java או MinTime בשפת C# , המקבלת את המערך arr , ומספר שלם של ברגים – amount .

הפעולה תחזיר את מספר השניות המינימלי הנדרש כדי שהמפעל יוכל לייצר לפחות amount ברגים סך הכול (מכל המכונות יחד).

אפשר להשתמש בפעולה שכתבתם בסעיף א.

לדוגמה: עבור המערך שבדוגמה שלעיל ו־ amount = 7 הפעולה תחזיר 6 .

הסבר: תפוקת המכונות בפרק זמן של 6 שניות מפורטת להלן:

המכונה באינדקס 0 מייצרת שישה ברגים,

המכונה באינדקס 1 אינה מייצרת אפילו בורג אחד,

והמכונה באינדקס 2 מייצרת שני ברגים.

סך הכול: שמונה ברגים (2 + 6).

כלומר, לשלוש המכונות נדרשות 6 שניות כדי לייצר לפחות שבעה ברגים סך הכול. בפחות מ־ 6 שניות המכונות אינן מספיקות לייצר שבעה ברגים (למשל בפרק זמן של 5 שניות שלוש המכונות מייצרות ביחד שישה ברגים סך הכול, כפי שמתואר בדוגמה בסעיף א).

/המשך בעמוד 5/

יש לענות על שלוש שאלות סך הכול מן הפרקים השני והשלישי (לכל שאלה – 25 נקודות).

פרק שני

שימו לב: בכל שאלה שנדרש בה מימוש אפשר להשתמש בפעולות של המחלקות: תור, מחסנית, עץ בינרי וחוליה, בלי לממש אותן. אם משתמשים בפעולות נוספות, יש לממש אותן.

4. "איבר קסם" הוא איבר בתור של מספרים שערכו שווה לסכום הערכים של האיבר שלפניו והאיבר שאחריו.

הערה: המספר הראשון בתור והמספר האחרון בתור אינם "איברי קסם".

א. כתבו פעולה ששמה isMagic בשפת Java או IsMagic בשפת C# , המקבלת תור – q מטיפוס שלם, ומספר שלם – m הגדול מ־ 0 וקטן או שווה לגודל התור.

הפעולה תחזיר true אם האיבר במקום ה־ m בתור הוא "איבר קסם", אחרת היא תחזיר false .

הערות: – בסיום הפעולה חובה לשמור על מבנה התור כפי שהתקבל.

– **אין** להשתמש בסעיף זה במערך או ברשימה מקושרת. פתרון הכולל שימוש בהם לא יזוכה בנקודות.

לדוגמה: עבור התור שלפניכם:

	ראש התור						סוף התור	
	1	*2*	*3*	*4*	*5*	*6*	*7*	
	5	11	6	9	3	6	3	

עבור $m = 1$ הפעולה תחזיר false (המספר הראשון בתור אינו "איבר קסם")

עבור $m = 2$ הפעולה תחזיר true $(5 + 6 = 11)$

עבור $m = 3$ הפעולה תחזיר false $(11 + 9 \neq 6)$

ב. כתבו פעולה ששמה nMagic בשפת Java או NMagic בשפת C# המקבלת תור מטיפוס שלם – q , ומספר שלם n הגדול מ־ 0 וקטן או שווה לגודל התור.

הפעולה תחזיר true אם כל האיברים הנמצאים במקומות שהם כפולה של n (המקום ה־ n בתור, המקום ה־ 2n בתור וכן הלאה בדילוגים של n מקומות) הם "איברי קסם". אחרת הפעולה תחזיר false .

אפשר להשתמש בפעולה שכתבתם בסעיף א.

הערות: – בפעולה זו אין צורך לשמור על התור שהתקבל.

– **אין** להשתמש בסעיף זה במערך או ברשימה מקושרת. פתרון הכולל שימוש בהם לא יזוכה בנקודות.

דוגמאות: עבור התור שבדוגמה שלעיל:

עבור $n = 2$ הפעולה תחזיר true מכיוון שכל האיברים הנמצאים במקומות שהם כפולה של 2 $(2, 4, 6)$ הם "איברי קסם".

עבור $n = 4$ הפעולה תחזיר true מכיוון שהאיבר במקום ה־ 4 הוא "איבר קסם" (אין בתור איברים נוספים במקומות שהם כפולה של 4).

עבור $n = 3$ הפעולה תחזיר false מכיוון שהאיבר במקום ה־ 3 אינו "איבר קסם".

/המשך בעמוד 6/

5. נתונה המחלקה **Patient** – חולה בחדר מיון, ולה שתי תכונות:

- id – מספר הזהות של החולה, מטיפוס שלם
- priority – רמת הדחיפות של הטיפול בחולה. רמת הדחיפות מיוצגת במספר מטיפוס שלם בין 1 ל־ 10 . ככל שהמספר גבוה יותר, רמת הדחיפות גבוהה יותר.

הניחו שלתכונות המחלקה יש פעולות get/Get ו־ set/Set .

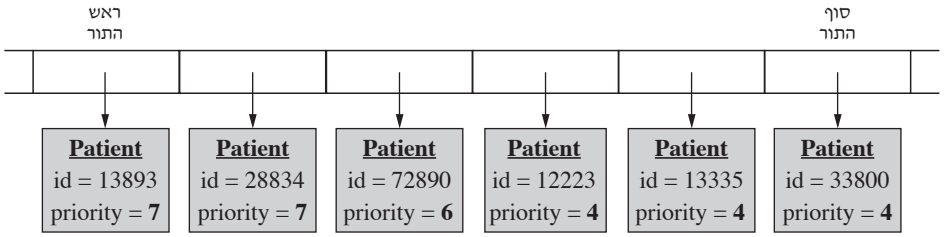
סדר הטיפול בחולים בחדר המיון מתנהל באופן שלהלן:

ככל שרמת הדחיפות של הטיפול גבוהה יותר, החולה מטופל מוקדם יותר. כאשר יש יותר מחולה אחד באותה רמת דחיפות, החולה שהגיע קודם לחדר מיון מטופל מוקדם יותר.

כדי לשמור על סדר הטיפול נבנתה המחלקה **PriorQueue** – תור עדיפויות, ולה תכונה אחת:

- q – הפניה לתור, מטיפוס Patient

דוגמה לתור q :



הניחו שהחולים שרמת הדחיפות שלהם זהה מוצגים בדוגמה לפי סדר הגעתם לחדר המיון.

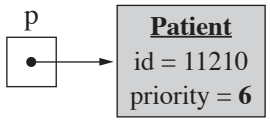
א. ממשו את הפעולה שלפניכם השייכת למשק המחלקה PriorQueue :

Java – public void priorityInsert (Patient p)

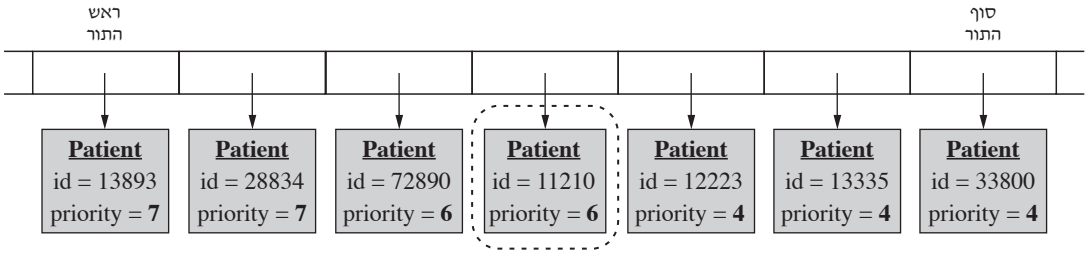
C# – public void PriorityInsert (Patient p)

הפעולה מקבלת חולה חדש – p ומכניסה אותו לתור q בהתאם לכללים של חדר המיון הכתובים לעיל.

לדוגמה: עבור התור המוצג לעיל והעצם שלפניכם:



התור ייראה כך לאחר ההכנסה:



(שימו לב: המשך השאלה בעמוד הבא.)

ב. מדי פעם רמת הדחיפות של חולה מסוים משתנה במהלך שהותו בחדר המיון.

ממשו את הפעולה שלפניכם השייכת לממשק המחלקה PriorQueue :

Java – public void update (int id, int pri)

C# – public void Update (int id, int pri)

הפעולה מקבלת מספר זהות של חולה הנמצא בתור – id , ומספר – pri המייצג את רמת הדחיפות המעודכנת שלו. הפעולה תעדכן את התכונה priority של החולה ותמקם אותו בתור לטיפול במקום המתאים לו (בהתאם לרמת הדחיפות המעודכנת – pri). אם בתור כבר יש חולים אחרים באותה רמת דחיפות – pri , החולה הנוכחי – id יוצב אחריהם כאילו הגיע אחריהם לחדר המיון.

/המשך בעמוד 8/

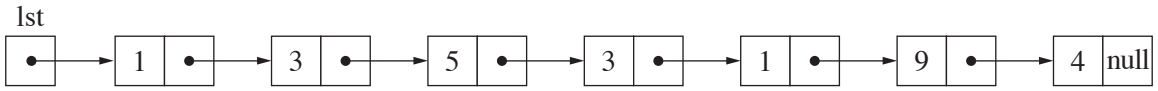
6. שימו לב: לשאלה זו שני נוסחים: בשפת Java בעמוד 8 ובשפת C# בעמוד 9.

לפותרים בשפת Java

א. נתונה הפעולה what :

```
public static Node<Integer> what (Node<Integer>lst, int x)
{
    if (lst == null)
        return null;
    Node<Integer> temp = what (lst.getNext(),x);
    if (lst.getValue() == x)
        return temp;
    lst.setNext (temp);
    return lst;
}
```

נתונה שרשרת חוליות – lst מטיפוס שלם:



(1) עקבו אחר הפעולה what (lst, 1) , והציגו את השרשרת שהפעולה מחזירה.

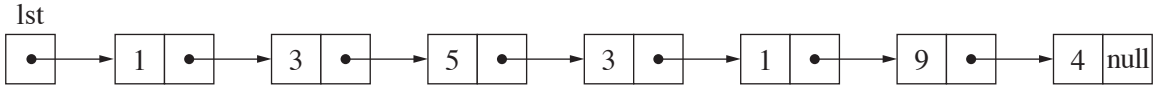
(2) מה עושה הפעולה what ? הסבירו את תשובתכם.

(3) מהי סיבוכיות הפעולה what ? נמקו את תשובתכם.

ב. נתונה הפעולה guess :

```
public static void guess (Node<Integer>lst)
{
    if (lst != null)  {
        Node<Integer> temp = what (lst.getNext(), lst.getValue());
        lst.setNext (temp);
        guess (lst.getNext());
    }
}
```

נתונה שרשרת חוליות – lst , מטיפוס שלם:



(1) עקבו אחר הפעולה guess (lst) , והציגו את השרשרת lst בסיום הפעולה.

בסעיף זה אין צורך לעקוב אחר הפעולה what .

(2) מה עושה הפעולה guess ? הסבירו את תשובתכם.

(3) מהי סיבוכיות הפעולה guess ? נמקו את תשובתכם.

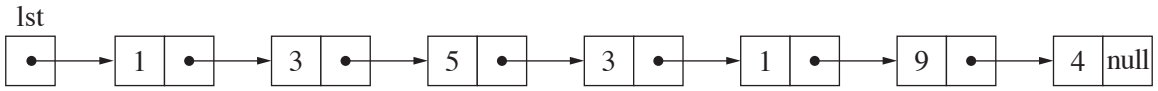
/המשך בעמוד 9/

לפותרים בשפת C#

א. נתונה הפעולה What :

```
public static Node<int> What (Node<int>lst, int x)
{
    if (lst == null)
        return null;
    Node<int> temp = What (lst.GetNext(),x);
    if (lst.GetValue() == x)
        return temp;
    lst.SetNext (temp);
    return lst;
}
```

נתונה שרשרת חוליות – lst מטיפוס שלם:



(1) עקבו אחר הפעולה What (lst, 1) , והציגו את השרשרת שהפעולה מחזירה.

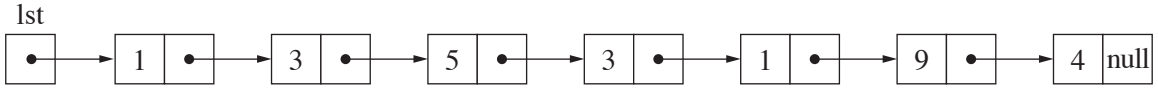
(2) מה עושה הפעולה What ? הסבירו את תשובתכם.

(3) מהי סיבוכיות הפעולה What ? נמקו את תשובתכם.

ב. נתונה הפעולה Guess :

```
public static void Guess (Node<int>lst)
{
    if (lst != null)  {
        Node<int> temp = What (lst.GetNext(), lst.GetValue());
        lst.SetNext (temp);
        Guess (lst.GetNext());
    }
}
```

נתונה שרשרת חוליות – lst מטיפוס שלם:



(1) עקבו אחר הפעולה Guess (lst) , והציגו את השרשרת lst בסיום הפעולה.

בסעיף זה אין צורך לעקוב אחר הפעולה What .

(2) מה עושה הפעולה Guess ? הסבירו את תשובתכם.

(3) מהי סיבוכיות הפעולה Guess ? נמקו את תשובתכם.

/המשך בעמוד 10/

7. 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 למחלקה 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 ב**כל** התחנות בעיר (כל קו כזה יופיע פעם אחת בלבד בשרשרת).

הערה: אין חשיבות לסדר הקווים בשרשרת.

/המשך בעמוד 11/

פרק שלישי

בפרק זה שאלות בשלושה מסלולים:

אלגוריתמים, עמודים 11–12.

מודלים חישוביים, עמודים 13–14.

תכנות מונחה עצמים בשפת Java , עמודים 16–19; תכנות מונחה עצמים בשפת C# , עמודים 20–23.

בחרו בשאלות מתוך מסלול אחד בלבד.

אלגוריתמים

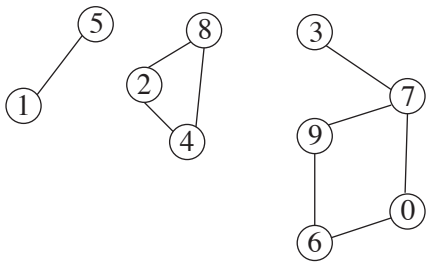
8. נתון גרף, לא קשיר ולא מכוון $G(V, E)$ שבו n קודקודים, מ־ v_0 עד v_{n-1} .

א. כתבו אלגוריתם המוצא ומחזיר את כל הקודקודים שיש מסלול בין קודקוד בגרף – v_j וביניהם.

הערה: יש לכתוב אלגוריתם יעיל שאינו עובר על כל המסלולים האפשריים.

דוגמה: עבור הגרף שלפניכם, וקודקוד 3 , האלגוריתם יחזיר את הקודקודים 0 , 6 , 7 , 9 .

$G(V, E)$

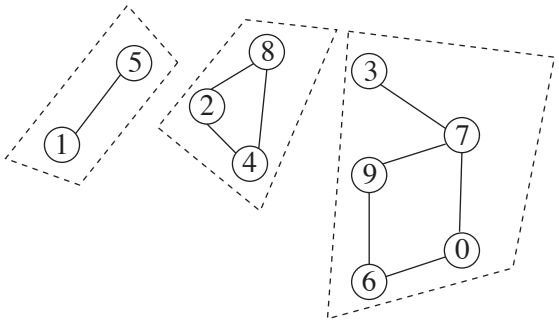


הסבר: קיים מסלול בין הקודקוד 3 ובין הקודקודים 0 , 6 , 7 , 9 .

ב. "רכיב קשירות" בגרף לא מכוון $G(V, E)$ הוא קבוצת קודקודים שבה בין כל שני קודקודים יש מסלול, ואין שום קשת היוצאת מקודקוד בקבוצה לקודקוד שאינו בקבוצה. קודקוד שאין קשת בינו ובין שום קודקוד אחר יהיה בקבוצה משלו.

דוגמה: עבור הגרף שבדוגמה לעיל, שלושת רכיבי הקשירות מסומנים בקו מקווקו.

$G(V, E)$



כתבו אלגוריתם המוצא ומחזיר את רכיב הקשירות הקטן ביותר (כלומר את הקבוצה שבה המספר המינימלי של קודקודים) בגרף $G(V, E)$.

למשל עבור הדוגמה שלעיל, האלגוריתם יחזיר את הקודקודים 1 , 5 .

הניחו שיש רק רכיב קשירות אחד שהוא הקטן ביותר.

הערה: יש לכתוב אלגוריתם יעיל שאינו עובר על כל המסלולים האפשריים.

/המשך בעמוד 12/

9. לפניכם שש טענות א–ו. בחרו בחמש מהן, וציינו בנוגע לכל טענה שבחרתם אם היא נכונה או לא נכונה.

אם הטענה נכונה – נמקו מדוע, ואם הטענה לא נכונה – הביאו דוגמה נגדית.

א. כל עץ המתקבל מהרצת DFS על גרף G לא מכוון, יכול להתקבל גם מהרצת BFS על אותו הגרף.

ב. כל עץ המתקבל מהרצת DFS על גרף G לא מכוון מלא הוא עץ שבו לכל צומת יש רק בן אחד.

ג. נתון גרף מכוון G וקודקוד v. אם אפשר להגיע מקודקוד v לכל אחד מן הקודקודים בגרף, ואפשר גם להגיע מכל אחד מן הקודקודים אל קודקוד v, הגרף G הוא בהכרח גרף קשיר היטב (חזק).

ד. אם בגרף G שאינו מכוון המסלול הקצר ביותר מן הצומת s_j אל הצומת s_n הוא $S[s_j \ldots \mathbf{s_m} \ldots \mathbf{s_k} \ldots s_n]$, בהכרח התת־מסלול מ־ s_m ועד s_k הוא המסלול הקצר ביותר מן הצומת s_m אל הצומת s_k.

ה. גרף שיש בו מעגל אינו יכול להיות גרף דו־צדדי.

ו. לכל גרף G ממושקל, לא מכוון, יש עץ פורש מינימלי יחיד.

/המשך בעמוד 13/

מודלים חישוביים

10. בשאלה זו שני סעיפים, א–ב, שאין ביניהם קשר. ענו על שני הסעיפים.

א. לפניכם שתי טענות (1)–(2) בנוגע לשפות L_1 , L_2 שמעל הא"ב $\{a, b\}$.

בעבור כל טענה, ציינו אם היא נכונה או לא נכונה. אם הטענה נכונה – נמקו מדוע,

ואם היא לא נכונה – הביאו דוגמה נגדית. אין קשר בין הטענות.

(1) אם L_1 , L_2 הן שפות לא רגולריות, בהכרח $L_1 \cap L_2$ היא שפה לא רגולרית.

(2) $L_1^n \cdot L_2^n$ תמיד שווה ל־ $(L_1 \cdot L_2)^n$.

ב. נתונה השפה L מעל הא"ב $\{a, b, c\}$:

$L = \{w \mid w$ אינם מופיעים ב־ w רצפים ab , bc , $\#_a(w) + \#_c(w) =$ מספר זוגי $\}$

$\#_a(w)$ מציין את מספר המופעים של a במילה w .

$\#_c(w)$ מציין את מספר המופעים של c במילה w .

דוגמה למילה בשפה L היא cba , כי סכום המופעים של האות a (1) והאות c (1) הוא מספר זוגי (2) ,

והרצפים ab ו־ bc אינם מופיעים במילה.

(1) לפניכם חמש מילים. בנוגע לכל אחת מהן ציינו אם המילה שייכת לשפה L או לא. נמקו את תשובתכם.

$aaccc$, aab , ε , baa , $cbbba$.

(2) בנו אוטומט סופי דטרמיניסטי שאינו מלא המקבל את השפה L .

/המשך בעמוד 14/

11. בשאלה זו שני סעיפים, א–ב, שאין ביניהם קשר. ענו על שני הסעיפים.

א. לפניכם שש שפות מעל האו"ב $\{0,1\}$:

Σ^* – מציין את שפת כל המילים מעל האו"ב $\{0,1\}$.

$L_1 = \emptyset$ $\quad L_4 = \{0110\}$

$L_2 = \Sigma^*$ $\quad L_5 = \{\varepsilon, 110, 00, 001\}$

$L_3 = \{\varepsilon\}$ $\quad L_6 = \{1, 0110, 110, 01\}$

כתבו את השפה המתקבלת מכל אחת מחמש הפעולות שלהלן:

(1) $L_5 \cap L_6$

(2) L_2^R

(3) $L_1 \cdot L_6$

(4) $L_3 \cdot L_4$

(5) $L_4 \cdot L_5$

ב. נתונה השפה L מעל האו"ב $\{a, b, c\}$:

$$L = \left\{ (ab)^k \, c^m \, b^{m+3k} \mid k, m >= 0 \right\}$$

בנו אוטומט מחסנית דטרמיניסטי המקבל את השפה L.

/המשך בעמוד 15/

<u>שימו לב</u>: המשך המבחן בעמוד הבא.

תכנות מונחה עצמים בשפת Java

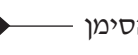
12. בחברה להשכרת כלֵי רכב "סעו לשלום" פותחה מערכת ממוחשבת שבה המחלקות האלה:

Contract – חוזה, **Vehicle** – כלִי רכב, **Car** – מכונית, **Truck** – משאית, **Motorcycle** – אופנוע.

להלן פירוט תכונות המחלקות:

- למחלקה חוזה (Contract) שלוש תכונות: name – שם לקוח (מחרוזת), days – מספר ימי השכרה (מטיפוס שלם), kilo – מספר הקילומטרים שנסע הלקוח (מטיפוס שלם).
- למחלקה כלִי רכב (Vehicle) שתי תכונות: מזהה כלי רכב – id (מחרוזת), חוזה – contract (Contract).
- למחלקה מכונית (Car) שלוש תכונות: מזהה כלי רכב – id (מחרוזת), חוזה – contract (Contract), מספר מקומות ישיבה – seats (מטיפוס שלם).
- למחלקה משאית (Truck) שלוש תכונות: מזהה כלי רכב – id (מחרוזת), חוזה – contract (Contract), משקל מקסימלי להעמסה – max (מטיפוס שלם).
- למחלקה אופנוע (Motorcycle) שלוש תכונות: מזהה כלי רכב – id (מחרוזת), חוזה – contract (Contract), אופנוע שטח – offRoad (בוליאני. אם האופנוע הוא אופנוע שטח התכונה היא אמת ואם לא, היא שקר).

א. **(1)** סרטטו תרשים היירכייה המתאר את הקשר בין המחלקות של המערכת הממוחשבת.

יש לסמן ירושה באמצעות החץ  והכלה באמצעות הסימן  .

(2) כתבו את כותרות המחלקות ואת התכונות שלהן. הניחו שהפעולות get ו־ set קיימות בכל התכונות של המחלקות, ואין צורך לממש אותן.

ב. נתונה הפעולה הבונה של המחלקה Contract :

```
public Contract (String name, int days, int kilo)
```

אין צורך לממש את הפעולה.

(1) לפניכם כותרת הפעולה הבונה של המחלקה Vehicle . הפעולה מקבלת שם לקוח, מספר ימי השכרה, מספר קילומטרים שנסע הלקוח ומזהה כלי הרכב.

```
public Vehicle (String name, int days, int kilo, String id)
```

ממשו את הפעולה הבונה.

(2) לפניכם כותרת הפעולה הבונה של המחלקה Car . הפעולה מקבלת שם לקוח, מספר ימי השכרה, מספר קילומטרים שנסע הלקוח, מזהה כלי הרכב ומספר מקומות ישיבה.

```
public Car (String name, int days, int kilo, String id, int seats)
```

ממשו את הפעולה הבונה.

(שימו לב: המשך השאלה בעמוד הבא.)

התעריף הבסיסי שהחברה גובה על השכרת כלי רכב – Vehicle הוא 60 שקלים ליום השכרה ו־ 2 שקלים לכל קילומטר של נסיעה.

מחיר ההשכרה של מכונית (Car) הוא לפי התעריף הבסיסי.

מחיר ההשכרה של משאית (Truck) הוא לפי התעריף הבסיסי ונוסף על כך סכום חד־פעמי של 500 שקלים.

מחיר ההשכרה של אופנוע (Motorcycle) הוא מחצית מן התעריף הבסיסי.

ג. הפעולה payment מחזירה מספר ממשי השווה לסכום שהלקוח נדרש לשלם בעבור כל אחד מסוגי כלי הרכב שהחברה משכירה (בהתאם לעצם שזימן את הפעולה).

(1) כתבו את הפעולה payment במחלקה Vehicle (כאמור לעיל, הסכום לתשלום הוא לפי התעריף הבסיסי).

(2) הוסיפו את הפעולה payment במחלקה/ות האחרות כדי לבצע את הנדרש (רק במחלקות שיש בהן צורך), לפי העקרונות של תכנות מונחה עצמים.

<u>הערה</u>: **אין** להשתמש בפעולה instanceOf בסעיף זה ובפעולות של המחלקה Object ואין לשנות את תכונות המחלקות. פתרון הכולל שימושים כאלה לא יזוכה בנקודות.

/המשך בעמוד 18/

13. לפניכם כותרות המחלקות AA , BB והתכונות שלהן.

הפעולות הבונות של המחלקות מסומנות ב- *** (תיתכן יותר מפעולה בונה אחת למחלקה).

```
public class AA
{
    private int x;
    ***
}
public class BB extends AA
{
    private AA f;
    ***
}
```

<u>הערה</u>: שימו לב – לשתי המחלקות אין פעולות get ו- set .

לפניכם המחלקה Test , הכוללת פעולה ראשית:

```
public class Test{
    public static void main(String[] args) {
        AA a1 = new AA ();
        AA a2  = new AA (8);
        AA a3 = new AA (3);
        AA a4 = new AA (a1);
        AA a5 = new AA (a2);
        BB b1 = new BB ();
        BB b2 = new BB (6, a4);
        BB b3 = new BB (5, a2);
        BB b4 = new BB (a4);
        BB b5 = new BB (a2);
    }
}
```

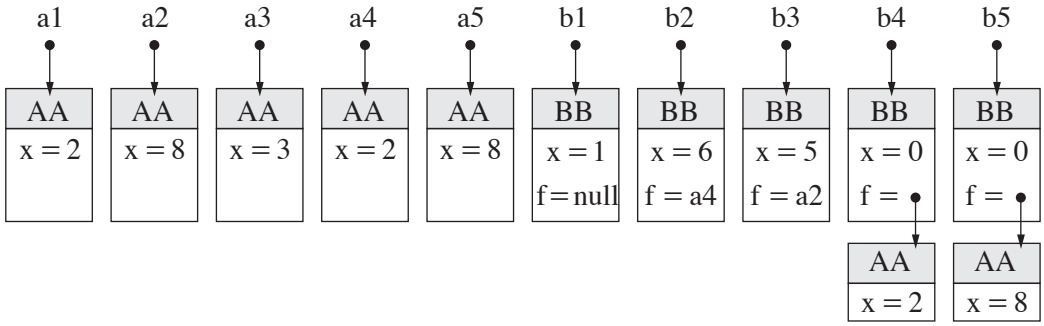
(שימו לב: המשך השאלה בעמוד הבא.)

לפניכם תרשים של עצמים שנוצרו בעקבות הרצת קטע הקוד.

כתבו במחלקות AA ו- BB את הפעולות הבונות הנדרשות כדי לקבל את העצמים שבתרשים.

ציינו בעבור כל אחד מן העצמים שנוצרו את הפעולה הבונה המתאימה לו.

הערה: **<u>אין</u>** להוסיף פעולות שאינן בונות במחלקות AA ו- BB . פתרון המוסיף פעולות שאינן בונות לא יזוכה בנקודות.



/המשך בעמוד 20/

תכנות מונחה עצמים בשפת C#

14. בחברה להשכרת כלֵי רכב "סעו לשלום" פותחה מערכת ממוחשבת שבה המחלקות האלה:

Contract – חוזה, **Vehicle** – כלִי רכב, **Car** – מכונית, **Truck** – משאית, **Motorcycle** – אופנוע.

להלן פירוט תכונות המחלקות:

- למחלקה חוזה (Contract) שלוש תכונות: name – שם לקוח (מחרוזת), days – מספר ימי השכרה (מטיפוס שלם), kilo – מספר קילומטרים שנסע הלקוח (מטיפוס שלם).
- למחלקה כלִי רכב (Vehicle) שתי תכונות: מזהה כלי רכב – id (מחרוזת), חוזה – contract (Contract).
- למחלקה מכונית (Car) שלוש תכונות: מזהה כלי רכב – id (מחרוזת), חוזה – contract (Contract), מספר מקומות ישיבה – seats (מטיפוס שלם).
- למחלקה משאית (Truck) שלוש תכונות: מזהה כלי רכב – id (מחרוזת), חוזה – contract (Contract), משקל מקסימלי להעמסה – max (מטיפוס שלם).
- למחלקה אופנוע (Motorcycle) שלוש תכונות: מזהה כלי רכב – id (מחרוזת), חוזה – contract (Contract), אופנוע שטח – offRoad (בוליאני. אם האופנוע הוא אופנוע שטח התכונה היא אמת ואם לא, היא שקר).

א. **(1)** סרטטו תרשים היירכייה המתאר את הקשר בין המחלקות של המערכת הממוחשבת.

יש לסמן ירושה באמצעות החץ והכלה באמצעות הסימן  .

(2) כתבו את כותרות המחלקות ואת התכונות שלהן. הניחו שהפעולות Get ו־ Set קיימות בכל התכונות של המחלקות, ואין צורך לממש אותן.

ב. נתונה הפעולה הבונה של המחלקה Contract :

```
public Contract (string name, int days, int kilo)
```

אין צורך לממש את הפעולה.

(1) לפניכם כותרת הפעולה הבונה של המחלקה Vehicle . הפעולה מקבלת שם לקוח, מספר ימי השכרה, מספר קילומטרים שנסע הלקוח ומזהה כלי הרכב.

```
public Vehicle (string name, int days, int kilo, string id)
```

ממשו את הפעולה הבונה.

(2) לפניכם כותרת הפעולה הבונה של המחלקה Car . הפעולה מקבלת שם לקוח, מספר ימי השכרה, מספר קילומטרים שנסע הלקוח, מזהה כלי הרכב ומספר מקומות ישיבה.

```
public Car (string name, int days, int kilo, string id, int seats)
```

ממשו את הפעולה הבונה.

(שימו לב: המשך השאלה בעמוד הבא.)

התעריף הבסיסי שהחברה גובה בעבור השכרת כלִי רכב – Vehicle הוא 60 שקלים ליום השכרה ו־ 2 שקלים לכל קילומטר של נסיעה.

מחיר ההשכרה של מכונית (Car) הוא לפי התעריף הבסיסי.

מחיר ההשכרה של משאית (Truck) הוא לפי התעריף הבסיסי ונוסף על כך סכום חד־פעמי של 500 שקלים.

מחיר ההשכרה של אופנוע (Motorcycle) הוא מחצית מן התעריף הבסיסי.

ג. הפעולה Payment מחזירה מספר ממשי השווה לסכום שהלקוח נדרש לשלם בעבור כל אחד מסוגי כלי הרכב שהחברה משכירה (בהתאם לעצם שזימן את הפעולה).

(1) כתבו את הפעולה Payment במחלקה Vehicle (כאמור לעיל, הסכום לתשלום הוא לפי התעריף הבסיסי).

(2) הוסיפו את הפעולה Payment במחלקה/ות האחרות כדי לבצע את הנדרש (רק במחלקות שיש בהן צורך), לפי העקרונות של תכנות מונחה עצמים.

<u>הערה</u>: **אין** להשתמש בפעולות is ו־ as בסעיף זה ובפעולות של המחלקה Object ואין לשנות את תכונות המחלקות. פתרון הכולל שימושים כאלה לא יזוכה בנקודות.

/המשך בעמוד 22/

15. לפניכם כותרות המחלקות AA , BB והתכונות שלהן.

הפעולות הבונות של המחלקות מסומנות ב- *** (תיתכן יותר מפעולה בונה אחת למחלקה).

```
public class AA
{
    private int x;
    ***
}
public class BB : AA
{
    private AA f;
    ***
}
```

<u>הערה</u>: שימו לב – לשתי המחלקות אין פעולות Get ו- Set .

לפניכם המחלקה Test , הכוללת פעולה ראשית:

```
public class Test{
    public static void Main(string[] args) {
        AA a1 = new AA ();
        AA a2  = new AA (8);
        AA a3 = new AA (3);
        AA a4 = new AA (a1);
        AA a5 = new AA (a2);
        BB b1 = new BB ();
        BB b2 = new BB (6, a4);
        BB b3 = new BB (5, a2);
        BB b4 = new BB (a4);
        BB b5 = new BB (a2);
    }
}
```

(שימו לב: המשך השאלה בעמוד הבא.)

לפניכם תרשים של עצמים שנוצרו בעקבות הרצת קטע הקוד.

כתבו במחלקות AA ו־ BB את הפעולות הבונות הנדרשות כדי לקבל את העצמים שבתרשים.

ציינו בעבור כל אחד מן העצמים שנוצרו את הפעולה הבונה המתאימה לו.

הערה: **<u>אין</u>** להוסיף פעולות שאינן בונות במחלקות AA ו־ BB . פתרון המוסיף פעולות שאינן בונות לא יזוכה בנקודות.