

מדעי המחשב

הוראות לנבחן

א. משך הבחינה: שלוש שעות.

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

פרק ראשון	—	$(10 \times 1) + (15 \times 1)$	—	25 נקודות
פרק שני	—	(25×2)	—	50 נקודות
פרק שלישי	—	(25×1)	—	25 נקודות
סך הכול	—		—	100 נקודות

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

ד. הוראות מיוחדות:

(1) את כל התוכניות שאתה נדרש לכתוב בשפת מחשב בפרקים הראשון והשני

כתוב בשפה אחת בלבד — Java או C#.

(2) רשום על הכריכה החיצונית של המחברת באיזו שפה אתה כותב — Java או C#.

(3) רשום על הכריכה החיצונית של המחברת את שם המסלול שלמדת.

המסלול הוא אחד מארבעת המסלולים האלה:

מערכות מחשב ואסמבלי, מבוא לחקר ביצועים, מודלים חישוביים, תכנות מונחה עצמים.

הערה: בתוכניות שאתה כותב לא יורדו לך נקודות, אם תכתוב אות גדולה במקום אות קטנה או להפך.

כתוב במחברת הבחינה בלבד. רשום "טיוטה" בראש כל עמוד המשמש טיוטה.

כתיבת טיוטה בדפים שאינם במחברת הבחינה עלולה לגרום לפסילת הבחינה.

ההנחיות בשאלון זה מנוסחות בלשון זכר ומכוונות לנבחנות ולנבחנים כאחד.

בהצלחה!

שים לב: השאלות מתחילות בעמוד 3.

השאלות

בשאלון זה שלושה פרקים.

עליך לענות על שאלות משלושת הפרקים, לפי ההוראות בכל פרק.

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

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

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

פרק ראשון (25 נקודות)

ענה על שאלה 1 – חובה (10 נקודות).

1. כתוב פעולה חיצונית בשפת Java או בשפת C#, המקבלת מערך arr מטיפוס שלם ומספר num מטיפוס שלם. הפעולה תחזיר את האינדקס הראשון של התא שסכום המספרים מתחילת המערך עד אליו (כולל) גדול מן המספר num. אם אין תא כזה, תחזיר הפעולה -1.

דוגמה: במערך arr שלפניך, בעבור num שווה ל-9, תחזיר הפעולה את המספר 4, משום שהוא האינדקס הראשון של התא שסכום האיברים מתחילת המערך עד אליו (כולל) גדול מן המספר 9.

0	1	2	3	4	5
3	-2	6	2	1	3

דוגמה נוספת: במערך arr שלפניך, בעבור num שווה ל-11, תחזיר הפעולה את המספר -1, משום שאין תא שסכום המספרים מתחילת המערך עד אליו (כולל) גדול מן המספר 11.

0	1	2	3	4	5
1	2	-2	0	7	3

ענה על אחת מן השאלות 2-3 (15 נקודות).

2. נתונה המחלקה **Weight** – משקל, שיש לה שתי תכונות:

kilo – מספר קילוגרמים, המקבלת ערכים מ־0 ומעלה, מטיפוס שלם.

gram – מספר גרמים, המקבלת ערכים מ־0 עד 999 (כולל), מטיפוס שלם.

הנח שלכל תכונה במחלקה הוגדרו בשפת Java פעולות get ו־set ובשפת C# פעולות Get ו־Set.

לפניך ממשק המחלקה **Weight** הכתוב בשפת Java ובשפת C#:

תיאור הפעולה	כותרת הפעולה
פעולה בונה המאתחלת את התכונות kilo ו־ gram לאפס.	public Weight ()
פעולה בונה המקבלת ערכים בקילוגרמים וגרמים. הנח שמספר הקילוגרמים במשתנה kilo גדול או שווה ל־0, ומספר הגרמים במשתנה gram הוא בין 0 ל־999 (כולל).	public Weight (int kilo, int gram)
פעולה בונה המקבלת ערך בגרמים. מספר הגרמים במשתנה totalGram יכול להיות מעל 999. הפעולה תחשב ותכניס את מספר הקילוגרמים ל־ kilo ואת מספר הגרמים ל־ gram. <u>דוגמה:</u> אם מספר הגרמים במשתנה totalGram הוא 3,215 התכונה kilo תקבל 3 והתכונה gram תקבל 215. הנח שמספר הגרמים במשתנה totalGram גדול או שווה ל־0.	public Weight (int totalGram)
פעולה המקבלת עצם אחר מטיפוס Weight ומוסיפה את ערכיו למשקל של העצם הנוכחי. הנח ש־ other אינו null.	בשפת Java – public void add (Weight other) בשפת C# – public void Add (Weight other)
פעולה המקבלת עצם אחר מטיפוס Weight ומחזירה true אם משקל העצם הנוכחי קטן מן המשקל האחר. הנח ש־ other אינו null.	בשפת Java – public boolean less (Weight other) בשפת C# – public bool Less (Weight other)

(שים לב: המשך השאלה בעמוד הבא.)

- א. מְמַשׁ את שתי הפעולות הבונות של המחלקה **Weight** : `public Weight (int kilo, int gram)` ו- `public Weight (int totalGram)` (אין צורך לממש את הפעולה הבונה ללא פרמטרים).
- ב. מְמַשׁ את הפעולות `add` ו- `less` בשפת Java, או את הפעולות `Add` ו- `Less` בשפת C#. **Weight**.
- ג. נתונה המחלקה **AllWeights** שיש לה תכונה אחת: `arr` – מערך מטיפוס **Weight**. לפניך פעולה של ממשק המחלקה **AllWeights** הכתובה בשפת Java ובשפת C#:

כותרת הפעולה	תיאור הפעולה
<code>public Weight sum ()</code> – בשפת Java	פעולה המחזירה עצם מטיפוס <code>Weight</code> השווה לסכום המשקלים של איברי המערך. <u>הערה</u> : אין לשנות את איברי המערך.
<code>public Weight Sum ()</code> – בשפת C#	

מְמַשׁ את הפעולה `sum` בשפת Java או את הפעולה `Sum` בשפת C#.

הנח שהתאים במערך אינם `null`.

הערה: תוכל להשתמש בכל אחת מן הפעולות של ממשק המחלקה **Weight**.

3. באכסניית הנוער "חלומות טובים" יש 3 קומות, 1-3. באכסניה 200 חדרים סך הכול. באכסניה שני סוגי חדרים: חדר זוגי וחדר יחיד. המחיר ללילה בחדר יחיד הוא 50 שקלים ובחדר זוגי 100 שקלים. במערכת המחשוב של האכסניה מוגדרת המחלקה **Room** – חדר, שיש לה שלוש תכונות:
- roomNum – מספר חדר בן שלוש ספרות, מטיפוס שלם. הספרה הראשונה (משמאל) מייצגת את מספר הקומה שבה החדר נמצא. לדוגמה: חדר מספר 231 נמצא בקומה מספר 2, וחדר 145 נמצא בקומה מספר 1.
 - roomType – סוג חדר: 1 הוא חדר יחיד, ו-2 הוא חדר זוגי, מטיפוס שלם.
 - nightsReserved – מספר הלילות שהחדר מוזמן מעכשיו, מטיפוס שלם. אם nightsReserved שווה ל-0 – החדר פנוי, ואם הוא מעל 0 – החדר תפוס.
- הנח שלכל תכונה במחלקה הוגדרו בשפת Java פעולות get ו-set ובשפת C# פעולות Get ו-Set.
- א. כתוב פעולה income בשפת Java או Income בשפת C#, במחלקה **Room**, המחזירה את ההכנסה המתקבלת מן החדר. חישוב ההכנסה הוא מספר הלילות המוזמנים כפול מחיר החדר.
- ב. במערכת המחשוב של האכסניה מוגדרת מחלקה נוספת, **Hostel** – אכסניה, שיש לה תכונה אחת: allRooms – מערך בגודל 200 מטיפוס **Room**. (החדרים במערך אינם ממוינים לפי סדר כלשהו). הנח שהתאים במערך אינם null.

לפניך ממשק חלקי של המחלקה **Hostel** הכתוב בשפת Java ובשפת C#:

תיאור הפעולה	כותרת הפעולה
<p>הפעולה מוצאת את החדר הראשון במערך allRooms שעונה על התנאים האלה:</p> <p>פנוי ומן הסוג type.</p> <p>הפעולה מעדכנת בחדר את מספר הלילות המבוקש המופיע במשתנה nights ומחזירה את מספר החדר. אם אין במערך חדר פנוי מן הסוג המבוקש, הפעולה מחזירה -1.</p>	<p>בשפת Java – public int orderRoom (int type, int nights)</p> <p>בשפת C# – public int OrderRoom (int type, int nights)</p>
<p>הפעולה מחזירה מערך בגודל 3 שבו מופיעים סך כל סכומי ההכנסה מכל החדרים בכל אחת מן הקומות באכסניה.</p> <p>באינדקס 0 מופיע סכום ההכנסה מקומה 1,</p> <p>באינדקס 1 מופיע סכום ההכנסה מקומה 2,</p> <p>ובאינדקס 2 מופיע סכום ההכנסה מקומה 3.</p>	<p>בשפת Java – public int [] floorIncome ()</p> <p>בשפת C# – public int [] FloorIncome ()</p>

(1) ממש את הפעולה orderRoom בשפת Java או את הפעולה OrderRoom בשפת C#, במחלקה **Hostel**.

(2) ממש את הפעולה floorIncome בשפת Java או את הפעולה FloorIncome בשפת C#,

במחלקה **Hostel**.

חובה להשתמש בפעולה שכתבת בסעיף א.

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 מן השאלות 4-6 (לכל שאלה – 25 נקודות).

4. **א.** כתוב פעולה חיצונית isExist בשפת Java או IsExist בשפת C#. הפעולה תקבל מספר num – מטיפוס שלם בין 0 ל-9 (כולל), ומחסנית stk – מטיפוס שלם. הפעולה תחזיר true אם יש במחסנית מספר שספרת האחדות שלו שווה למספר num, אחרת היא תחזיר false. הנח שהמספרים במחסנית stk אינם שליליים.

דוגמה: עבור num שווה ל-8 והמחסנית stk שלפניך:



הפעולה תחזיר true, משום שבמחסנית יש מספר 568 שספרת האחדות שלו היא 8. הערה: חובה לשמור על מבנה המחסנית עם סיום הפעולה.

ב. לשם פתרון סעיף זה בלבד תוכל להשתמש בפעולה שלפניך ללא צורך לממש:

כותרת הפעולה	תיאור הפעולה
בשפת Java – <code>public static Stack <Integer> clone (Stack <Integer> s)</code>	הפעולה מקבלת מחסנית מטיפוס שלם ומחזירה העתק מדויק של המחסנית בלי לשנות את המחסנית המקורית.
בשפת C# – <code>public static Stack <int> Clone (Stack <int> s)</code>	

נגזר: הספרה המשמעותית במספר היא הספרה הכי שמאלית שלו.

לדוגמה הספרה 3 היא הספרה המשמעותית במספר 32, והספרה 5 היא הספרה המשמעותית במספר 541.

כתוב פעולה חיצונית allExist בשפת Java או AllExist בשפת C#, המקבלת מחסנית stk שאינה ריקה, מטיפוס שלם. הפעולה תחזיר true אם כל הספרות המשמעותיות במספרים שבמחסנית מופיעות בספרת האחדות במספרים כלשהם במחסנית, אחרת היא תחזיר false. הנח שהמספרים במחסנית stk אינם שליליים.

דוגמה: עבור המחסנית `stk` שלפניך:

122	← ראש המחסנית
251	
565	
12334	
28	
7	

הפעולה תחזיר `true` כי כל הספרות המשמעותיות במספרים שבמחסנית – 1, 2, 5, 7 – מופיעות בספרת האחדות במספרים שבמחסנית.

דוגמה נוספת: עבור המחסנית `stk` שלפניך:

1223	← ראש המחסנית
245	
521	
12334	

הפעולה תחזיר `false` כי מתוך הספרות המשמעותיות שבמחסנית – 1, 2, 5 – הספרה 2 אינה מופיעה בספרת האחדות בשום מספר מן המספרים שבמחסנית.
הערה: אפשר להשתמש בפעולה שכתבת בסעיף א.

5. לקראת תחרות ריצת מרתון הוגדרה מחלקה **Competitor** המייצגת מתחרה שסיים את המסלול.

למחלקה יש שלוש תכונות:

- minutes – מספר הדקות שנדרשו למתחרה לסיים את המסלול (מספר הדקות אינו מוגבל ל- 60), מטיפוס שלם.
- seconds – מספר השניות שנדרשו למתחרה לסיים את המסלול (מספר השניות הוא עד 59 כולל), מטיפוס שלם.
- name – שם המתחרה מטיפוס מחרוזת.

הנח שלכל תכונה הוגדרו בשפת Java הפעולות get ו-set ובשפת C# הפעולות Get ו-Set.

נוסף על כך הוגדרה מחלקה בשם **Race** המאגדת אוסף של כל המתחרים שסיימו את המסלול. מספר המסיימים אינו ידוע.

הנח שאין שני מתחרים שסיימו את המסלול בזמן זהה.

לפניך ממשק חלקי של המחלקה **Race** הכתוב בשפת Java ובשפת C#:

סיבוכיות	תיאור הפעולה	כותרת הפעולה
$O(n)$	הפעולה מקבלת עצם מטיפוס Competitor ומוסיפה אותו לאוסף. <u>שים לב</u> : הפעולה הבאה של הממשק (rank בשפת Java או Rank בשפת C#) ודרישות הסיבוכיות שלה רלוונטיות לאופן שבו מממשים פעולה זו.	בשפת Java – <code>public void add (Competitor x)</code> בשפת C# – <code>public void Add (Competitor x)</code>
$O(n)$	הפעולה מקבלת דירוג ומחזירה את שם המתחרה, שלו דירוג זה באוסף. <u>הדרכה</u> : 1 הוא המתחרה שסיים את המסלול בזמן <u>הקצר ביותר</u> , 2 הוא המתחרה שסיים את המסלול בזמן השני הקצר ביותר וכן הלאה. הנח שקיים מתחרה בדירוג המבוקש. <u>הערה</u> : אין למחוק איברים מהאוסף.	בשפת Java – <code>public String rank (int x)</code> בשפת C# – <code>public string Rank (int x)</code>

(שים לב: המשך השאלה בעמוד הבא.)

שם לב:

- לכל פעולה יש הנחיות סיבוכיות זמן ריצה.
- n הוא מספר המתחרים באוסף.
- **חובה לעמוד בדרישות הסיבוכיות.**

בעבור הסעיפים א-ב עליך להשתמש במבנה נתונים מתאים העומד בדרישות השאלה.
תוכל להשתמש בכל מבנה נתונים שלמדת.

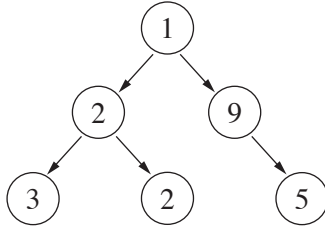
אפשר לכתוב פעולות במחלקה **Competitor**.

א. כתוב את תכונות המחלקה **Race**.

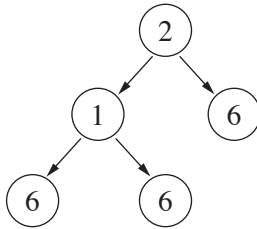
ב. ממש את פעולות המחלקה **Race** המופיעות בממשק המחלקה שבשאלה.

6. נגדיר: "עץ מספרים" הוא עץ בינארי מטיפוס שלם, שכל צומת בו מכיל ספרה בין 1 ל-9 (כולל), וכל מסלול בעץ מן השורש לעלה מייצג מספר: העלה מייצג את ספרת האחדות, הרמה שמעליו את ספרת העשרות וכן הלאה עד השורש של העץ.

דוגמה: בעץ המספרים שלפניך מיוצגים המספרים: 123, 122, 195 (במסלולים בעץ משמאל לימין).



דוגמה נוספת: בעץ המספרים שלפניך מיוצגים המספרים: 216, 216, 26 (במסלולים בעץ משמאל לימין).



כתוב פעולה חיצונית printAll בשפת Java או PrintAll בשפת C#. הפעולה תקבל עץ מספרים tree מטיפוס שלם ותדפיס את כל המספרים שהמסלולים בעץ מייצגים.
 אם tree הוא null הפעולה לא תדפיס דבר.
הערה: אין חשיבות לסדר שבו המספרים מודפסים.

פרק שלישי (25 נקודות)

בפרק זה שאלות בארבעה מסלולים:
 מערכות מחשב ואסמבלי, עמודים 13-15.
 מבוא לחקר ביצועים, עמודים 16-20.
 מודלים חישוביים, עמוד 21.
 תכנות מונחה עצמים בשפת Java, עמודים 22-25; תכנות מונחה עצמים בשפת C#, עמודים 26-29.
ענה על שאלה אחת במסלול שלמדת.

מערכות מחשב ואסמבלי

אם למדת מסלול זה, ענה על אחת מן השאלות 7-8 (25 נקודות).

7. בשאלה זו שלושה סעיפים א-ג. אין קשר בין הסעיפים. עליך לענות על כולם.
 א. לפניך קטע תוכנית באסמבלי.

```
MOV BX, 50H
MOV CL, 8
L1: MOV AX, [BX]
    ROL AX, CL
    MOV [BX], AX
    ADD BX, 2
    CMP BX, 57H
    JBE L1
```

לפניך מפת תאי הזיכרון מ-50H עד 57H, לפני ביצוע קטע התוכנית:

57H	56H	55H	54H	53H	52H	51H	50H	כתובת התא
88H	77H	66H	55H	44H	33H	22H	11H	תוכן התא

- הערה: שים לב שיש הבדל בין העברת נתונים מתאי זיכרון לאוגר של 8 סיביות ובין העברת נתונים מתאי זיכרון לאוגר של 16 סיביות.
- (1) עקוב בעזרת טבלת מעקב אחר ביצוע קטע התוכנית. בטבלת המעקב יש לכלול עמודה לכל אחד מן האוגרים המופיעים בקטע התוכנית. נוסף על כך יש לסרטט מפת זיכרון מתאימה.
- (2) הסבר מה מבצע קטע התוכנית.

(שים לב: המשך השאלה בעמוד הבא.)

ב. לפניך קטע תוכנית הכתוב בשפות Java ו- C#. כתוב קטע מתאים בשפת אסמבלי.

```
while ( a > 0 || b <= c )
{
    a = b + c ;
    c -- ;
}
```

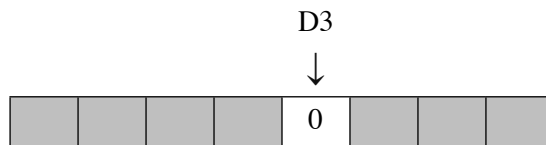
הנח שהמשתנים a, b, c מכוונים (signed) ומאוחסנים באוגרים AX, BX, CX בהתאם.

ג. לפניך שני היגדים בנוגע לאוגר AH. בעבור כל אחד מן ההיגדים, קבע אם הוא נכון או לא נכון. אם ההיגד נכון נמק את תשובתך, ואם ההיגד לא נכון, הצב באוגר AH מספר המוכיח שהיגד זה אינו מתקיים.

i. בהנחה שבסיבית D3 (הסיבית הרביעית מצד ימין) של האוגר AH מאוחסן הערך 0, שתי הפקודות 1 ו-2 שלפניך מבצעות את אותה הפעולה. כלומר הערך של האוגר AH זהה לאחר הרצת שתי הפקודות.

פקודה 2	פקודה 1
SUB AH, 8	AND AH, 11110111b

היעזר באיור שלפניך:



ii. בהנחה שבסיבית D3 (הסיבית הרביעית מצד ימין) של האוגר AH מאוחסן הערך 1, שתי הפקודות 1 ו-2 שלפניך מבצעות את אותה הפעולה. כלומר הערך של האוגר AH זהה לאחר הרצת שתי הפקודות.

פקודה 2	פקודה 1
SUB AH, 8	AND AH, 11110111b

היעזר באיור שלפניך:



8. בשאלה זו שני סעיפים א-ב. אין קשר בין הסעיפים. עליך לענות על שניהם.

א. לפניך קטע תוכנית באסמבלי.

MOV AX, 620H

SUB AH, 76H

קבע מה הם הערכים של הדגלים ZF, CF, SF לאחר ביצוע קטע התוכנית. הסבר את קביעתך.

ב. במקטע הנתונים הוגדר הנתון שלפניך:

ARR DB 10 DUP (?)

כתוב שגרה (פרוצדורה) בשם TEST המקבלת דרך מחסנית שני מספרים שלמים. המספר הראשון שמועבר במחסנית הוא ערך של איבר המופיע במערך ARR, והמספר השני המועבר במחסנית הוא האינדקס של אותו האיבר.

אם מתחילת המערך עד האינדקס שהועבר (לא כולל) יש לפחות ערך אחד השווה לערך של האיבר שהועבר, תאחסן השגרה (הפרוצדורה) את המספר 1 באוגר AL, אחרת – היא תאחסן 0 באוגר AL.

דוגמה: במערך ARR שלפניך מועברים הערך 5 והאינדקס 4.

index	0	1	2	3	4	5	6	7	8	9
value	3	16	0	3	5	3	5	16	5	0

בדוגמה זו תאחסן השגרה את המספר 0 באוגר AL, מכיוון שמתחילת המערך ועד האינדקס 4, הערך 5 אינו קיים.

דוגמה נוספת: במערך ARR שלפניך מועברים הערך 5 והאינדקס 8.

index	0	1	2	3	4	5	6	7	8	9
value	3	16	0	3	5	3	5	16	5	0

בדוגמה זו תאחסן השגרה את המספר 1 באוגר AL, מכיוון שמתחילת המערך ועד האינדקס 8, הערך 5 קיים.

מבוא לחקר ביצועים

אם למדת מסלול זה, ענה על אחת מן השאלות 9-10 (25 נקודות).

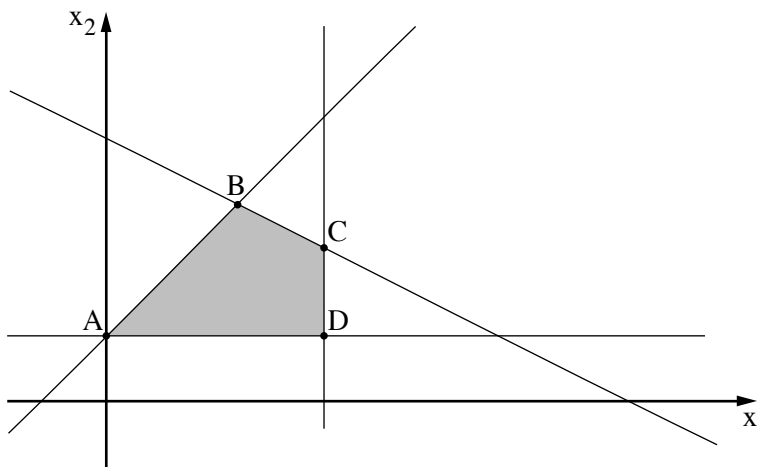
9. נתונה בעיית תכנון לינארי:

$$\max \{z = ax_1 + 10x_2\}$$

בכפוף לאילוצים האלה:

- (1) $x_1 + 3 \geq x_2$
- (2) $x_2 \geq 3$
- (3) $-0.5x_1 + 12 \geq x_2$
- (4) $x_1 \leq 10$

לפניך סרטוט של תחום הפתרונות האפשריים של הבעיה הנתונה.



(שים לב: המשך השאלה בעמוד הבא.)

- א. מצא את שיעורי הנקודות A, B, C, D .
- ב. לאורך הקטע BC הערך של פונקציית המטרה זהה. חשב את a .
- ג. לפניך ארבעה היגדים:
- יש רק פתרון אופטימלי אחד.
 - יש אין-ספור פתרונות אופטימליים.
 - הפתרון האופטימלי אינו חסום.
 - אין פתרון אופטימלי.
- קבע איזה מן ההיגדים i - iv נכון. העתק אותו למחברתך ונמק את קביעתך.
- ד. מבטלים את האילוץ השני $x_2 \geq 3$.
- (1) סרטט מחדש את תחום הפתרונות האפשריים.
 - (2) האם הפתרון האופטימלי השתנה לאחר שינוי זה? נמק את תשובתך.
 - (3) משנים גם את פונקציית המטרה למינימום $\min \{z = ax_1 + 10x_2\}$. האם הפתרון האופטימלי השתנה לאחר שינוי זה? נמק את תשובתך.

10. בשאלה זו שני סעיפים, א-ב. אין קשר בין הסעיפים. עליך לענות על שניהם.

- א. (1) בטבלה שלפניך נתון פתרון לבעיית תובלה. האם פתרון זה הוא פתרון בסיסי אפשרי לפי שיטת הפינה הצפונית-מערבית? נמק את תשובתך.

מקורות	יעדים				היצע
	A	B	C	D	
1	5 100	3	8 100	7 100	300
2	9	5 300	10 100	12 100	500
3	8	6	7	3 200	200
ביקוש	100	300	200	400	

(שים לב: המשך סעיף א של השאלה בעמוד הבא.)

(2) בטבלה שלפניך נתון פתרון בסיסי אפשרי שהתקבל לאחר k איטרציות בעבור בעיית תובלה, ונתון הערך של v_2 .

מקורות	יעדים			היצע	u_i
	1	2	3		
A	10	3	8	100	
B	9	7	6	200	
C	18	9	4	400	
D	11	6	8	300	
ביקוש	150	500	350		
v_j		0			

- i. העתק את הטבלה למחברתך והשלם בה את הערכים $u_1, u_2, u_3, u_4, v_1, v_3$.
- ii. הסבר מדוע הפתרון אינו אופטימלי.
- iii. עליך לבצע איטרציה נוספת, כלומר $k + 1$. מהו המשתנה שיוצא מן הבסיס באיטרציה זו?
- iv. סרטט במחברתך טבלה חדשה ורשום בה את הפתרון שהתקבל לאחר איטרציה זו.
- v. האם פחתה העלות הכוללת של בעיית התובלה?

(שים לב: סעיף ב של השאלה בעמוד הבא.)

ב. לפניך 6 ערים a, b, c, d, e, f המחוברות ברשת של כבישים. העלות הכוללת של הנסיעה בכביש מורכבת משני רכיבים: עלות הדלק ואגרת הנסיעה.

לפניך מטריצת סמיכויות של עלות הדלק בשקלים לנסיעה בין כל שתי ערים.

	a	b	c	d	e	f
a	0	6	3	∞	∞	∞
b	6	0	7	2	11	∞
c	3	7	0	10	2	2
d	∞	2	10	0	8	2
e	∞	11	2	8	0	6
f	∞	∞	2	2	6	0

לפניך מטריצת סמיכויות של אגרת הנסיעה בשקלים לנסיעה בין כל שתי ערים.

	a	b	c	d	e	f
a	0	4	20	∞	∞	∞
b	4	0	2	9	1	∞
c	20	2	0	10	13	5
d	∞	9	10	0	3	13
e	∞	1	13	3	0	3
f	∞	∞	5	13	3	0

מצא את המסלולים הזולים ביותר לנסיעה לפי האלגוריתם של דיקסטר, מעיר a לכל אחת מן הערים a, b, c, d, e, f . רשום את עלות הנסיעה הכוללת הנמוכה ביותר מעיר a לכל אחת מן הערים האחרות.

מודלים חישוביים

אם למדת מסלול זה, ענה על אחת מן השאלות 11-12 (25 נקודות).

11. בנה מכונת טיורינג המקבלת כקלט את המספרים x, y, z , מתחילת הסרט (המספרים אינם שליליים).

x הוא המספר הראשון (מתחילת הסרט), y הוא השני ו- z הוא השלישי.

כל מספר כתוב בצורה אונרית.

הסימן # מפריד בין מספר למספר.

דוגמה: אם הקלט מקבל 2 עבור x , 3 עבור y ו-1 עבור z הסרט ייראה כך:

	1	1	#	1	1	1	#	1	△	△	△
--	---	---	---	---	---	---	---	---	---	---	---	------

דוגמה נוספת: אם הקלט מקבל 0 עבור x , 1 עבור y ו-0 עבור z הסרט ייראה כך:

	#	1	#	△	△	△
--	---	---	---	---	---	---	------

המכונה תחשב את ערך הפונקציה שלפניך:

$$f(x, y, z) = \begin{cases} y + z & x = 0 \\ x + y & x > 0 \text{ וגם } x \text{ זוגי} \\ x & x \text{ אי-זוגי} \end{cases}$$

הפלט ייכתב על הסרט במקום כלשהו כערך אונרי בין שני סימני \$.

12. א. בנה אוטומט מחסנית עבור השפה L_1 מעל הא"ב $\{a,b,c\}$ המורכב מרצפים מן הצורה $a^n b^k c^n$ כך ש- n הוא מספר אי-זוגי ושארית החלוקה של k בשלוש היא אחת. בין כל שני רצפים מפרידה האות b .

דוגמאות למילים ששייכות לשפה L_1 :

abbbbcbbaabcccc, abc

דוגמאות למילים שאינן שייכות לשפה L_1 :

abbc — כי מספר הפעמים שהאות b מופיעה הוא 2, ושארית החלוקה של 2 ב-3 היא 2.

abcabc — כי האות b אינה מפרידה בין שני הרצפים.

abccc — כי מספר הפעמים שהאות a מופיעה אינו שווה למספר הפעמים שהאות c מופיעה.

aabcc — כי מספר הפעמים שהאות a והאות c מופיעות הוא זוגי.

ב. נתונה השפה L_2 מעל הא"ב $\{a,b,c\}$

$$L_2 = \{a^k b^m c^x \mid 0 \leq k < 5, 0 \leq m < 5, 0 \leq x\}$$

נגדיר $L_3 = L_2 \cap L_1$.

כתוב את כל המילים שבשפה L_3 .

תכנות מונחה עצמים בשפת Java

אם למדת מסלול זה ואתה כותב בשפת Java, ענה על אחת מן השאלות 13-14 (25 נקודות).

13. נתונות המחלקות: **A, B, C, D, E**.

```
public class A
{
    protected int x ;
    public A ()
    {
        this.x = 9 ;
        System.out.println ( "A. x = " +this.x ) ;
    }
    public A ( int x )
    {
        this.x = x ;
        System.out.println ( "A. x = " + this.x ) ;
    }
    public int getX () { return this.x ; }
    public int foo () { return this.x ; }
}

public class B extends A
{
    public B () { super () ; }
    public B ( int x ) { super ( x ) ; }
    public int foo () { return this.x +1 ; }
}

public class C extends B
{
    public C () { super () ; }
    public C ( int x ) { super ( x ) ; }
    public int foo () { return this.x + 2 ; }
    public int bar () { return this.x ; }
}
```

```
public class D extends C
{
    public D ()
    {
        super () ;
        this.x ++ ;
        System.out.println ( "D. x = " + this.x ) ;
    }
    public D ( int x )
    {
        super ( x ) ;
        System.out.println ( "D. x = " + this.x ) ;
    }
    public D ( int x, int y )
    {
        super ();
        this.x = this.x + x + y;
        System.out.println ( "D. x = " + this.x ) ;
    }
    public int foo () { return this.x - 1 ; }
}

public class E extends C
{
    public E () { super () ; }
    public int bar () { return this.x +1 ; }
}
```

(שים לב: המשך השאלה בעמוד הבא.)

- א. סרטט תרשים הייררכייה בין המחלקות A , B , C , D , E . יש לסמן ירושה ממחלקה באמצעות החץ —→ .
- ב. לפניך כותרת הפעולה:

```
public static int getType ( Object m )
```

הפעולה מחזירה 1 אם m מטיפוס A , 2 אם m מטיפוס B , 3 אם m מטיפוס C , 4 אם m מטיפוס D ו-5 אם m מטיפוס E .
ממש את הפעולה.

הדרכה: כדי לבדוק את סוג העצם יש להיעזר בפעולות getX , bar , foo .

אין להשתמש בפעולה instanceof ובפעולות של המחלקה Object , ואין לשנות את המחלקות A , B , C , D , E .
הנח ש- m שייך לאחת מן המחלקות A , B , C , D , E ואינו null .

- ג. לפניך המחלקה **Tester** :

```
public class Tester
```

```
{
    public static void main ( String [] args )

    {
        A a1 = new B () ;
        A a2 = new E () ;
        A a3 = new D () ;
        A a4 = new D ( 5 ) ;
        A a5 = new D ( 3, 7 ) ;
    }
}
```

כתוב את הפלט של הפעולה main .

14. בחברת התקשורת "אוזניים לעתיד" יש 4 סוגי עובדים. כל עובד שייך לסוג אחד בלבד.

עובד רגיל – עובד רגיל בחברה.

חבר ועד – עובד שחבר בוועד העובדים.

טכנאי – עובד שאחראי על כמה מחשבים בחברה (כל טכנאי אחראי על מספר שונה של מחשבים).

אחראי פרויקט – עובד שמנהל פרויקטים מיוחדים בחברה. הוא אחראי על 15 עובדים ובהם עובדים רגילים, חברי ועד וטכנאים.

החלטות מתקבלות באמצעות הצבעה בהשתתפות כל עובדי החברה. ההצבעה נערכת באמצעות נקודות. לכל עובד בחברה יש מספר נקודות שונה, לפי סוגו.

לפניך טבלה ובה נתונים בנוגע למספר הנקודות שיש לכל עובד לפי סוגו:

סוג העובד	מספר הנקודות
עובד רגיל	מקבל 4 נקודות עבור היותו עובד בחברה, ונוסף על כך הוא מקבל עוד נקודה עבור כל שנת ותק. <u>דוגמה:</u> לעובד רגיל שעובד 5 שנים בחברה יש 9 נקודות: 4 נקודות עבור היותו עובד בחברה ועוד 5 נקודות – נקודה עבור כל שנת ותק בחברה.
חבר ועד	מקבל פי 2 ממספר הנקודות שעובד רגיל מקבל, ונוסף על כך הוא מקבל 2 נקודות עבור כל שנה שבה הוא חבר בוועד. <u>דוגמה:</u> לחבר ועד שעובד 5 שנים בחברה ומתוכן הוא חבר בוועד 3 שנים יש 24 נקודות: 18 נקודות – פי 2 מעובד רגיל שעובד 5 שנים בחברה ועוד 6 נקודות – 2 נקודות עבור כל שנה שהוא חבר בוועד.
טכנאי	מקבל נקודות כמו עובד רגיל, נוסף על כך הוא מקבל נקודה עבור כל מחשב שבאחריותו. <u>דוגמה:</u> לטכנאי שעובד 5 שנים בחברה ואחראי על 4 מחשבים יש 13 נקודות: 9 נקודות – כפי שמקבל עובד רגיל שעובד 5 שנים בחברה ועוד 4 נקודות – נקודה עבור כל מחשב שבאחריותו.
אחראי פרויקט	מקבל נקודות כמו עובד רגיל, ונוסף על כך הוא מקבל מספר נקודות השווה לסך הנקודות של כל 15 העובדים שעליהם הוא אחראי. <u>דוגמה:</u> לאחראי פרויקט שעובד 5 שנים בחברה, וסך כל הנקודות של 15 העובדים שבאחריותו הוא 80, יש 89 נקודות: 9 נקודות – כפי שמקבל עובד רגיל שעובד 5 שנים בחברה ועוד 80 נקודות – סך כל הנקודות של 15 העובדים שבאחריותו.

(שים לב: המשך השאלה בעמוד הבא.)

בחברה רוצים לבנות מערכת ממוחשבת שתציג את מספר הנקודות של כל עובד בחברה. לשם כך הגדירו 4 מחלקות: **Employee** – עובד רגיל, **UnionMember** – חבר ועד, **Technician** – טכנאי, **Supervisor** – אחראי פרויקט. כל עובד שייך למחלקה אחת בלבד.

את הסעיפים שלפניך יש לפתור באופן המתאים ביותר לעקרונות תכנות מונחה עצמים.

א. סרטט תרשים הייררכייה בין המחלקות. יש לסמן ירושה ממחלקה באמצעות החץ \longrightarrow והכלה באמצעות הסימן \blacklozenge .

ב. עבור כל אחת מן המחלקות כתוב את כותרת המחלקה, את התכונות שלה ואת הפעולה `getScore()` המחזירה את מספר הנקודות בהצבעה. אין צורך לכתוב פעולה בונה ופעולות `get` ו-`set`.
הנחיה: הנתונים הקבועים של מספר הנקודות אינם חלק מתכונות המחלקות, ויש להשתמש בהם רק בכתיבת הפעולה `getScore`.

ג. הועלתה הצעה לשנות את שם החברה ל"תקשורת עתידית". כל העובדים הרגילים והטכנאים תמכו בהצעה וכל חברי הוועד והאחראים לפרויקטים התנגדו לה. רק אם סך הנקודות בעד ההצעה גדול מסך הנקודות נגד ההצעה – ההצעה מתקבלת, אחרת היא אינה מתקבלת.
נתונה כותרת פעולה המקבלת מערך של כל העובדים ומחזירה `true` אם ההצעה התקבלה, אחרת היא מחזירה `false`.

```
public static boolean isAccepted (Object [] arr)
```

מִמַּש את הפעולה.

תכנות מונחה עצמים בשפת C#

אם למדת מסלול זה ואתה כותב בשפת C#, ענה על אחת מן השאלות 15-16. (25 נקודות)

15. נתונות המחלקות: **A, B, C, D, E**.

```
public class A
{
    protected int x ;
    public A ()
    {
        this.x = 9 ;
        Console.WriteLine ( "A. x = " +this.x );
    }
    public A ( int x )
    {
        this.x = x ;
        Console.WriteLine ( "A. x = " + this.x );
    }
    public int GetX () { return this.x ; }
    public virtual int Foo () { return this.x ; }
}

public class B : A
{
    public B () : base () {}
    public B ( int x ) : base ( x ) {}
    public override int Foo () { return this.x +1 ; }
}

public class C : B
{
    public C () : base () {}
    public C ( int x ) : base ( x ) {}
    public override int Foo () { return this.x +2 ; }
    public virtual int Bar () { return this.x ; }
}
```

```
public class D : C
{
    public D () : base ()
    {
        this.x ++ ;
        Console.WriteLine ( "D. x = " + this.x );
    }
    public D ( int x ) : base ( x )
    {
        Console.WriteLine ( "D. x = " + this.x );
    }
    public D ( int x, int y ) : base ()
    {
        this.x = this.x + x + y;
        Console.WriteLine ( "D. x = " + this.x );
    }
    public override int Foo () { return this.x - 1 ; }
}

public class E : C
{
    public E () : base () {}
    public override int Bar () { return this.x +1 ; }
}
```

(שים לב: המשך השאלה בעמוד הבא.)

- א. סרטט תרשים הייררכייה בין המחלקות A , B , C , D , E . יש לסמן ירושה ממחלקה באמצעות החץ —▶ .
- ב. לפניך כותרת הפעולה:

```
public static int GetType ( Object m )
```

הפעולה מחזירה 1 אם m מטיפוס A , 2 אם m מטיפוס B , 3 אם m מטיפוס C , 4 אם m מטיפוס D ו- 5 אם m מטיפוס E .

ממש את הפעולה.

הדרכה: כדי לבדוק את סוג העצם יש להיעזר בפעולות GetX , Bar , Foo .

אין להשתמש בפעולות is ו- as ובפעולות של המחלקה Object , ואין לשנות את המחלקות A , B , C , D , E . הנחש m שייך לאחת מן המחלקות A , B , C , D , E ואינו null .

- ג. לפניך המחלקה **Tester** :

```
public class Tester
```

```
{
    public static void Main ( string [] args )
    {
        A a1 = new B () ;
        A a2 = new E () ;
        A a3 = new D () ;
        A a4 = new D ( 5 ) ;
        A a5 = new D ( 3, 7 ) ;
    }
}
```

כתוב את הפלט של הפעולה Main .

16. חברת התקשורת "אוזניים לעתיד" יש 4 סוגי עובדים. כל עובד שייך לסוג אחד בלבד.

עובד רגיל — עובד רגיל בחברה.

חבר ועד — עובד שחבר בוועד העובדים.

טכנאי — עובד שאחראי על כמה מחשבים בחברה (כל טכנאי אחראי על מספר שונה של מחשבים).

אחראי פרויקט — עובד שמנהל פרויקטים מיוחדים בחברה. הוא אחראי על 15 עובדים ובהם עובדים רגילים, חברי ועד וטכנאים.

החלטות מתקבלות באמצעות הצבעה בהשתתפות כל עובדי החברה. ההצבעה נערכת באמצעות נקודות. לכל עובד בחברה יש מספר נקודות שונה, לפי סוגו.



לפניך טבלה ובה נתונים בנוגע למספר הנקודות שיש לכל עובד לפי סוגו:

סוג העובד	מספר הנקודות
עובד רגיל	מקבל 4 נקודות עבור היותו עובד בחברה, ונוסף על כך הוא מקבל עוד נקודה עבור כל שנת ותק. <u>דוגמה:</u> לעובד רגיל שעובד 5 שנים בחברה יש 9 נקודות: 4 נקודות עבור היותו עובד בחברה ועוד 5 נקודות — נקודה עבור כל שנת ותק בחברה.
חבר ועד	מקבל פי 2 ממספר הנקודות שעובד רגיל מקבל, ונוסף על כך הוא מקבל 2 נקודות עבור כל שנה שבה הוא חבר בוועד. <u>דוגמה:</u> לחבר ועד שעובד 5 שנים בחברה ומתוכן הוא חבר בוועד 3 שנים יש 24 נקודות: 18 נקודות — פי 2 מעובד רגיל שעובד 5 שנים בחברה ועוד 6 נקודות — 2 נקודות עבור כל שנה שהוא חבר בוועד.
טכנאי	מקבל נקודות כמו עובד רגיל, נוסף על כך הוא מקבל נקודה עבור כל מחשב שבאחריותו. <u>דוגמה:</u> לטכנאי שעובד 5 שנים בחברה ואחראי על 4 מחשבים יש 13 נקודות: 9 נקודות — כפי שמקבל עובד רגיל שעובד 5 שנים בחברה ועוד 4 נקודות — נקודה עבור כל מחשב שבאחריותו.
אחראי פרויקט	מקבל נקודות כמו עובד רגיל, ונוסף על כך הוא מקבל מספר נקודות השווה לסך הנקודות של כל 15 העובדים שעליהם הוא אחראי. <u>דוגמה:</u> לאחראי פרויקט שעובד 5 שנים בחברה, וסך כל הנקודות של 15 העובדים שבאחריותו הוא 80, יש 89 נקודות: 9 נקודות — כפי שמקבל עובד רגיל שעובד 5 שנים בחברה ועוד 80 נקודות — סך כל הנקודות של 15 העובדים שבאחריותו.

(שים לב: המשך השאלה בעמוד הבא.)

בחברה רוצים לבנות מערכת ממוחשבת שתציג את מספר הנקודות של כל עובד בחברה. לשם כך הגדירו 4 מחלקות: **Employee** – עובד רגיל, **UnionMember** – חבר ועד, **Technician** – טכנאי, **Supervisor** – אחראי פרויקט. כל עובד שייך למחלקה אחת בלבד.

את הסעיפים שלפניך יש לפתור באופן המתאים ביותר לעקרונות תכנות מונחה עצמים.

א. סרטט תרשים הייררכייה בין המחלקות. יש לסמן ירושה ממחלקה באמצעות החץ  והכלה באמצעות הסימן  .

ב. עבור כל אחת מן המחלקות כתוב את כותרת המחלקה, את התכונות שלה ואת הפעולה () `GetScore` המחזירה את מספר הנקודות בהצבעה. אין צורך לכתוב פעולה בונה ופעולות `Get` ו-`Set`.

הנחיה: הנתונים הקבועים של מספר הנקודות אינם חלק מתכונות המחלקות, ויש להשתמש בהם רק בכתיבת הפעולה `GetScore`.

ג. הועלתה הצעה לשנות את שם החברה ל"תקשורת עתידית". כל העובדים הרגילים והטכנאים תמכו בהצעה וכל חברי הוועד והאחראים לפרויקטים התנגדו לה. רק אם סך הנקודות בעד ההצעה גדול מסך הנקודות נגד ההצעה – ההצעה מתקבלת, אחרת היא אינה מתקבלת. נתונה כותרת פעולה המקבלת מערך של כל העובדים ומחזירה `true` אם ההצעה התקבלה, אחרת היא מחזירה `false`.

```
public static bool IsAccepted (Object [] arr)
```

ממש את הפעולה.

בהצלחה!