

## מדעי המחשב ב'

2 יחידות לימוד (השלמה ל-5 יח"ל)

### הוראות לנבחן

- א. משך הבחינה: שלוש שעות.
- ב. מבנה השאלון ומפתח ההערכה: בשאלון זה שני פרקים.  
פרק ראשון – בפרק זה ארבע שאלות, — 50 נקודות  
ומהן יש לענות על שתיים.
- פרק שני – בפרק זה שאלות בחמישה מסלולים שונים. — 50 נקודות  
ענה על שאלות רק במסלול שלמדת,  
לפי ההוראות בקבוצת השאלות במסלול זה.
- סה"כ — 100 נקודות
- ג. חומר עזר מותר בשימוש: כל חומר עזר (חוץ ממחשב הניתן לתכנות).
- ד. הוראות מיוחדות:
- את כל התכניות שאתה נדרש לכתוב בשפת מחשב עילית בפרק הראשון כתוב בשפה אחת בלבד – פסקל אֲנֹ C.
  - רשום על הכריכה החיצונית של המחברת את שם המסלול שלמדת (אחד מחמשת המסלולים: מערכות מחשב ואסמבלר, תורת המחשב, מודלים חישוביים, חישוב מקבילי ומבוזר, תכנות מונחה עצמים).
  - רשום על הכריכה החיצונית של המחברת באיזו שפה אתה כותב – פסקל אֲנֹ C.

כתוב במחברת הבחינה בלבד, בעמודים נפרדים, כל מה שברצונך לכתוב בטיוטה (ראשי פרקים, חישובים וכדומה).  
רשום "טיוטה" בראש כל עמוד טיוטה. רישום טיוטות כלשהן על דפים שמחוץ למחברת הבחינה עלול לגרום לפסילת הבחינה!

ההנחיות בשאלון זה מנוסחות בלשון זכר ומכוונות לנבחנות ולנבחנים כאחד.

**ב ה צ ל ח ה !**

/המשך מעבר לדף/

## ה ש א ל ו ת

בשאלון זה שני פרקים: פרק ראשון ופרק שני.  
עליך לענות על שאלות משני הפרקים, לפי ההוראות בכל פרק.

### פרק ראשון (50 נקודות)

ענה על שתיים מהשאלות 4-1 (לכל שאלה – 25 נקודות).

1. לפניך אלגוריתם:

**סוד 1**  $(x, n)$

{הפעולה מקבלת מספרים שלמים  $n$  ו- $x$ , גדולים או שווים ל-0, ומחזירה מספר שלם}

(1) אם  $x = 0$  אז החזר 0

(2) אחרת החזר **סוד 1**  $(x - 1, n)$  + **סוד 2**  $(x - 1, n)$

**סוד 2**  $(x, n)$

{הפעולה מקבלת מספרים שלמים  $n$  ו- $x$ , גדולים או שווים ל-0, ומחזירה מספר שלם}

(1) אם  $x = 0$  אז החזר 1

(2) אחרת

(2.1) אם  $x > n$  אז

(2.1.1)  $x \leftarrow n$

(2.1.2) החזר **סוד 1**  $(x, n)$  + **סוד 2**  $(x, n)$

(2.2) אחרת החזר **סוד 1**  $(x - 1, n)$  + **סוד 2**  $(x - 1, n)$

א. מה יחזיר הזימון **סוד 2**  $(3, 10)$ ? רשום את המעקב אחר ביצוע האלגוריתם.

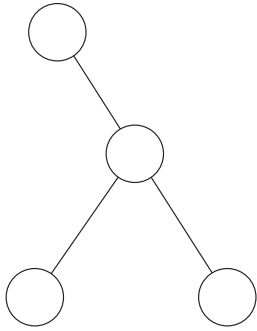
ב. מה יחזיר הזימון **סוד 1**  $(10, 10)$  + **סוד 2**  $(3, 10)$ ?

ג. נתון  $x_1 > n$  וגם  $x_2 > n$ .

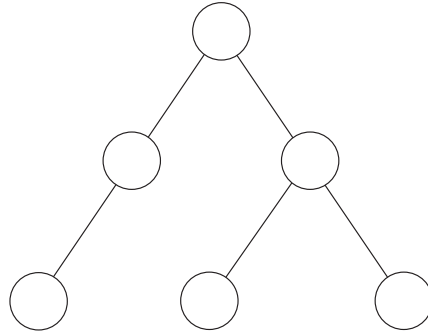
האם מתקיים: **סוד 2**  $(x_1, n)$  = **סוד 2**  $(x_2, n)$ ? נמק.

2. עץ בינרי T הוא עץ "פרו ורבו", אם קיים בו צומת אשר יש לו לפחות שני נכדים, וכל נכד מבין אחר. הערה: נכד הוא רק בן של בן.

דוגמה



לא עץ "פרו ורבו"



עץ "פרו ורבו"

כתוב אלגוריתם **האם\_פרו\_ורבו?**(T), המקבל עץ בינרי T ומחזיר 'אמת' אם הוא עץ "פרו ורבו", ואחרת – 'שקר'.

3.

במשחק קובייה "הטל וצבור" יש 15 סיבובים. במשחק משתתפים 17 שחקנים, וכל אחד מהם ממוספר במספר שלם בין 1 ל-17. במהלך המשחק כל שחקן מטיל קובייה. בהטלת הקובייה מתקבל מספר בין 1 ל-6.

בכל אחד מ-15 הסיבובים כל שחקן מבצע בתורו את ארבע הפעולות האלה:

- (i) בוחר מספר שלם, ומכריז מהו הערך שלו.
- (ii) מטיל את הקובייה 5 פעמים, ומסכם את המספרים שהתקבלו ב-5 ההטלות.
- (iii) מחשב את הערך המוחלט של ההפרש בין המספר שהכריז ב-(i) ובין הסכום שחישב ב-(ii).

(iv) מעדכן את מספר הנקודות הכולל שלו בדרך זו:

אם ההפרש שחישב ב-(iii) הוא 0, מספר הנקודות הכולל שלו מתאפס. אחרת – הוא מוסיף את הערך המוחלט של ההפרש למספר הנקודות הכולל שלו.

השחקן שצבר הכי פחות נקודות ב-15 סיבובים הוא המנצח.

אם בסיום המשחק יש ליותר משחקן אחד את המספר הכי נמוך של נקודות, אז מתבצע סיבוב נוסף. כל השחקנים משתתפים בסיבוב הנוסף, ומספר הנקודות של כל אחד מהם הוא מספר הנקודות שצבר מתחילת המשחק. כל עוד אין מנצח, יתקיים סיבוב נוסף. לפניך פעולות מתוך ממשק הטיפוס **משחק** בסביבת העבודה (כתובות בפסקל וב-C). הנח שקיימת הגדרה של טיפוס הנתונים `game_type`.

**ממשק בפסקל**

<p>הפעולה מחזירה משחק מאותחל. מספר הנקודות הכולל של כל משתתף מאותחל ל-0.</p>	<pre>procedure game_init   (var g: game_type);</pre>
<p>הפעולה מקבלת משחק <code>g</code> ונתוני שחקן עבור סיבוב אחד:  <code>num</code> – את מספר השחקן,  <code>announce</code> – את הערך שעליו הכריז השחקן,  <code>sum</code> – את סכום 5 ההטלות של השחקן בסיבוב זה. הפעולה מעדכנת את הניקוד הכולל של השחקן במשחק.  <b>הנחות:</b> <code>g</code> מאותחל,  <code>num</code> – שלם בין 1 ל-17,  <code>sum ≥ 5</code> , <code>announce ≥ 5</code>.</p>	<pre>procedure update_player_score   (var g: game_type;    num, announce, sum: integer);</pre>
<p>הפעולה מקבלת משחק <code>g</code>. אם המשחק הסתיים, הפעולה מחזירה את המספר של השחקן המנצח, אחרת – הפעולה מחזירה 0.</p>	<pre>function winner   (g: game_type): integer;</pre>
<p>הפעולה מחזירה את סכום 5 ההטלות של הקובייה.</p>	<pre>function throw_5_times: integer;</pre>

/המשך בעמוד 5/

**ממשק ב-C**

<p>הפעולה מחזירה משחק מאותחל. מספר הנקודות הכולל של כל משתתף מאותחל ל-0.</p>	<p>game_type game_init ()</p>
<p>הפעולה מקבלת משחק g ונתוני שחקן עבור סיבוב אחד:                  ב num – את מספר השחקן,                  ב announce – את הערך שעליו הכריז השחקן, ב sum – את סכום 5 ההטלות של השחקן בסיבוב זה. הפעולה מעדכנת את הניקוד הכולל של השחקן במשחק.                  הנחות: g מאותחל, num – שלם בין 1 ל-17, announce ≥ 5 , sum ≥ 5 .</p>	<p>void update_player_score (game_type *g, int num, int announce, int sum)</p>
<p>הפעולה מקבלת משחק g. אם המשחק הסתיים, הפעולה מחזירה את המספר של השחקן המנצח, אחרת – הפעולה מחזירה 0.</p>	<p>int winner (game type g)</p>
<p>הפעולה מחזירה את סכום 5 ההטלות של הקובייה.</p>	<p>int throw_5_times ()</p>

**א.** יצג את טיפוס הנתונים game\_type בסביבת העבודה.

**ב.** ממש בסביבת העבודה פעולה המקבלת משחק מאותחל, שבו מספר הנקודות ההתחלתי של כל משתתף הוא 0. הפעולה מריצה משחק, ומחזירה את המספר של השחקן המנצח.

4. נגדיר "בלוק" במחסנית כרצף של לפחות שני איברים זהים.

א. נתונה מחסנית M, שאיבריה הם מספרים שלמים.

כתוב תת-תכנית בסביבת העבודה שמקבלת את המחסנית M, ומחזירה מחסנית חדשה. עבור כל "בלוק" שבמחסנית M יש במחסנית החדשה איבר, שערכו הוא המכפלה של האיבר שב"בלוק" במספר הפעמים שהוא מופיע ב"בלוק".  
אין חשיבות לסדר האיברים במחסנית המוחזרת.

הערה: אין צורך לממש בסביבת העבודה את הפעולות של ממשק המחסנית.

דוגמה

המחסנית המוחזרת

-8
0
15

מחסנית M

5
5
5
0
0
0
0
0
3
2
-4
-4

ב. מהי סיבוכיות זמן הריצה של התת-תכנית שכתבת בסעיף א? הסבר.

הנח שסיבוכיות זמן הריצה של כל אחת מן הפעולות של ממשק המחסנית

היא  $O(1)$ .

**פרק שני (50 נקודות)**

בפרק זה שאלות בחמישה מסלולים שונים. ענה רק על שאלות במסלול שלמדת.

**מערכות מחשב ואסמבלר**

אם למדת מסלול זה, ענה על שתיים מהשאלות 5-8 (לכל שאלה – 25 נקודות).

5. יש לבנות מערכת חיבורים למחשב קיים, כגיבוי חירום לשמירת מספרי הטלפון של משטרת ישראל ושל מגן דוד אדום. במערכת יש רכיב זיכרון מסוג ROM עם שמונה כתובות של 8 סיביות כל אחת, וכן שני לחצני חירום. לחיצה על הלחצן הראשון תגרום לתוכנת המחשב להתקשר למשטרת ישראל, ולחיצה על הלחצן השני תגרום לתוכנת המחשב להתקשר למגן דוד אדום.

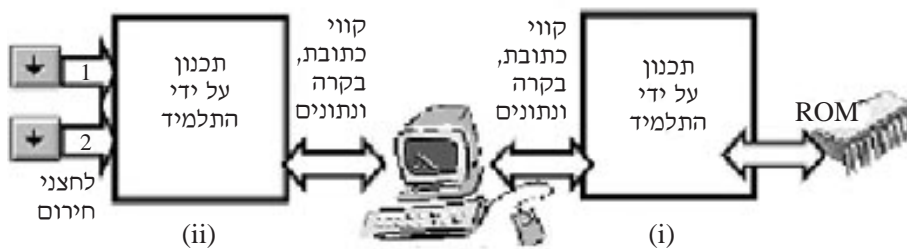
**א.** תכנן את מערכת החיבורים (ראה תרשים):

(i) לצורך חיבור זיכרון ה-ROM לקווי המחשב, החל מכתובת 0A520h .

(ii) לצורך חיבור שני הלחצנים לקווי המחשב, החל מכתובת 0F6h .

**ב.** סרטט את טבלת המיפוי של כתובות הקלט/פלט.

**ג.** סרטט את מערכת החיבורים למעבד לצורך חיבור רכיבי הזיכרון לכתובות המתקבלות ממפת הזיכרון.



6. בתחילת תכנית בשפת אסמבלר הוגדרו הנתונים בצורה זו:

ARR						FIRST		משתנה
00	01	02	03	04	05	06	07	כתובת
04	10	03	07	5B	10	0	0	נתון

שים לב: הנתונים והכתובות הם הקסדצימליים.

לפניך קטע התכנית:

```

                LEA     SI, ARR
                MOV     CX, 5
                MOV     AH, [SI]
                MOV     AL, AH
AGA1:          INC     SI
                CMP     AH, [SI]
                JLE     CONT1
                MOV     AH, [SI]
CONT1:         CMP     AL, [SI]
                JNB     CONT2
                MOV     AL, [SI]
CONT2:         LOOP   AGA1
                LEA     DI, FIRST
                MOV     [DI], AH
                MOV     [DI+1], AL
SOF:          NOP
    
```

א. עקוב בעזרת טבלת מעקב אחר הביצוע של קטע התכנית, ורשום מה יהיה התוכן של תאי הזיכרון שכתובותיהם מ-00 עד 07.

ב. מה מבצע קטע התכנית? בתשובתך התייחס לתוכן האוגרים AL ו-AH.

ג. החלף את ההוראה LOOP AGA1 בהוראה אחרת / הוראות אחרות, כך שקטע התכנית יבצע את הנדרש בלי להשתמש בהוראה LOOP.



7. סדרת פיבונצ'י היא סדרת מספרים, אשר החל מהאיבר השלישי כל איבר הוא סכום של

שני האיברים שלפניו. שני האיברים הראשונים בסדרה הם 0 ו-1:

0 1 1 2 3 5 8 13...

לפניך קטע של תכנית בשפת אסמבלר, המחשב את סדרת פיבונצ'י עד למספר בסדרה

שערכו 5 (כולל), ושומר את המספרים שהתקבלו במערך בזיכרון החל מכתובת BUFFER .

מקטע התכנית הושמטו פקודות ואופרנדים.

```

START:  MOV     SI, 2
        LEA     BX, [_____]
        MOV     AL, 0
        MOV     AH, 1
        MOV     [BX+SI-2], AL
        MOV     [BX+SI-1], AH
        INC     AL
        INC     AH
        MOV     [BX+SI], AL
        ADD     SI, 1
        MOV     DH, AH
        ADD     AH, AL
        MOV     AL, DH
        MOV     [BX+SI], [_____]
        [_____] [_____] , 1
        MOV     DH, AH
        ADD     [_____] , [_____]
        MOV     AL, DH
        MOV     [BX+SI], AL
        ADD     SI, 1
        MOV     [BX+SI], AH
SOF2:   NOP
    
```

העתק למחברתך את השורות שבהן הושמטו פקודות או אופרנדים, והשלם את החלקים החסרים כך שקטע התכנית יבצע את הנדרש.

8. לפניך שישה היגדים, (1)-(6).

קבע לכל אחד מהם אם הוא נכון או לא נכון, ונמק את תשובתך.

- (1) סדרת הוראות של קלט/פלט, הנדרשת במקרה של גישה ישירה לקבצים, זהה לסדרת הוראות של קלט/פלט, הנדרשת במקרה של גישה סדרתית.
- (2) במחשב הפועל עם שני מעבדים ניתן לשתף את זיכרון ה-RAM לשניהם.
- (3) במערכת שיתוף זמנים רק תהליך אחד נמצא בזיכרון הראשי בזמן מסוים.
- (4) במערכת רבת-מעבדים תהליך אינו יכול לעבוד עם יותר ממעבד אחד בו-זמנית.
- (5) במערכת רבת-תהליכים התהליך, שיש לו העדיפות הנמוכה ביותר מבין כל התהליכים הפעילים, לא יקבל הקצאת מעבד (מצב RUN), כל עוד לא הסתיימו התהליכים שיש להם עדיפות גבוהה יותר.
- (6) במערכת עם זיכרון וירטואלי ניתן להוציא תהליך, שאינו בביצוע, מחוץ לזיכרון הממשי.

**תורת המחשב**

אם למדת מסלול זה, ענה על שתיים מהשאלות 9-12 (לכל שאלה – 25 נקודות).

9. לפניך האלגוריתם **תעלומה**:

**תעלומה**  $(E, Y, N, X_0)$

$\{ E, Y, X_0 \}$  האלגוריתם מקבל מספר שלם  $N$  ושלושה מספרים ממשיים  $\{ E, Y, X_0 \}$

(1) אם  $Y = 0$

(1.1) אזי החזר 0

(1.2) אחרת

(1.2.1)  $F \leftarrow 0$

(1.2.2) כל עוד  $F = 0$  בצע

(1.2.2.1)  $X_2 \leftarrow X_0^{(N-1)}$

(1.2.2.2)  $X_1 \leftarrow ((N-1) * X_0 + Y / X_2) / N$

(1.2.2.3) אם  $E > |X_0 - X_1|$  אז  $F \leftarrow 1$

(1.2.2.4)  $X_0 \leftarrow X_1$

(1.2.3) החזר  $X_0$

א. עקוב בעזרת טבלת מעקב אחר ביצוע האלגוריתם **תעלומה**  $(0.01, 49, 2, 25)$ ,

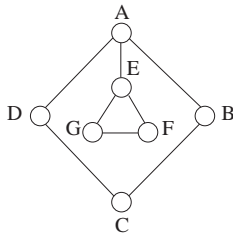
ורשום מה יוחזר.

ב. מה מבצע האלגוריתם **תעלומה**  $(E, Y, N, X_0)$  ?

10. יהיה גרף  $G$ , קשיר, לא מכוון, בלי קשתות מרובות (אין יותר מקשת אחת בין שני קדקודים) ובלי לולאות עצמיות. נגדיר "מסלול מעגלי" בגרף  $G$  כמסלול המתחיל ומסתיים באותו קדקוד, וכל קדקוד אחר במסלול מופיע בו רק פעם אחת. יהיה "כיסוי מעגלים" של גרף  $G$  אוסף של "מסלולים מעגליים" ב- $G$ , כך שכל קדקוד בגרף  $G$  נמצא ב"מסלול מעגלי" אחד בדיוק.

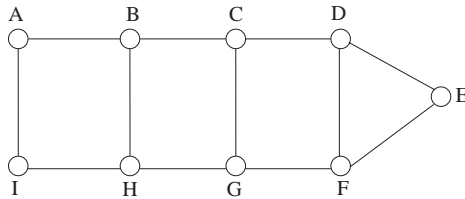
דוגמה

נתון הגרף  $G_1$ :



"כיסוי מעגלים" של הגרף  $G_1$  הוא אוסף המסלולים המעגליים:  $\{EFGE, ABCDA\}$ .

א. לפניך הגרף  $G_2$ :



- (i) רשום "כיסוי מעגלים" לגרף  $G_2$ .  
 (ii) האם הכיסוי שרשמת הוא "כיסוי מעגלים" יחיד? אם כן – נמק מדוע, אם לא – רשום "כיסוי מעגלים" נוסף.

- ב. תן דוגמה לגרף קשיר, לא מכוון, בלי קשתות מרובות ובלי לולאות עצמיות, שבו הדרגה של כל אחד מהקדקודים היא לפחות 2 ואין לו "כיסוי מעגלים".  
 ג. יהיה גרף  $G$ , קשיר, לא מכוון, בלי קשתות מרובות ובלי לולאות עצמיות. הגרף  $G'$  הוא גרף המכיל את כל הקדקודים של הגרף  $G$  ורק את הקשתות המופיעות ב"כיסוי מעגלים" שלו. נסח תנאי על הגרף  $G$  שיבטיח כי יש עבורו גרף  $G'$  קשיר.



**מודלים חישוביים**

אם למדת מסלול זה, ענה על שתיים מהשאלות 13-16 (לכל שאלה – 25 נקודות).

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

$$L = \left\{ \begin{array}{l} \text{אוסף כל המילים המתחילות ב-} b \text{ ומסתיימות ב-} b, \\ \text{יש בהן לפחות } c \text{ אחד, ומספר ה-} a \text{ בהן זוגי} \end{array} \right\}$$

בנה אוטומט סופי דטרמיניסטי, שיקבל את השפה  $L$ .

14. נגדיר פעולה חדשה על שלוש שפות  $L_C, L_B, L_A$  באופן הזה:

$L_A \# (L_B, L_C)$  היא קבוצת כל המילים הנמצאות ב-  $L_A$ , אך לא נמצאות ב-  $L_B$  וגם לא נמצאות ב-  $L_C$ .

א. לפניך שלוש השפות  $L_3-L_1$  מעל הא"ב  $\{a, b\}$ :

$$L_1 = \{ a^n b^m a^n \mid n, m > 0 \}$$

$$L_2 = \{ \text{מספר האותיות } a \text{ שונה ממספר האותיות } b \}$$

$$L_3 = \{ \text{מספר האותיות } b \text{ הוא אי-זוגי} \}$$

(i) תן דוגמה למילה באורך גדול מ-4 השייכת לשפה  $L_1 \# (L_2, L_3)$ .

הסבר מדוע המילה שייכת לשפה זו.

(ii) תן דוגמה למילה באורך 4 לפחות שאינה שייכת לשפה  $L_1 \# (L_2, L_3)$ .

הסבר מדוע המילה אינה שייכת לשפה זו.

(iii) תאר את אוסף המילים בשפה  $L_1 \# (L_2, L_3)$  מעל הא"ב  $\{a, b\}$ .

ב. נתון כי  $L_C, L_B, L_A$  הן שפות רגולריות.

הוכח שהשפה  $L_A \# (L_B, L_C)$  היא רגולרית.

15. לפניך שלוש השפות  $L_1, L_2, L_3$ :

$$L_1 = \{ a^n b^n \mid n > 0 \}$$

$$L_2 = \{ a^n b^n d a^k b^k \mid n, k \geq 0 \}$$

$$L_3 = \{ \text{כל המילים מעל הא"ב } \{a, b\} \text{ המכילות את הרצף } ba \}$$

לפניך חמש הטענות (i)-(v):

$$aaaabbbb \in L_1^2 \quad (i)$$

$$R(L_1) \cdot d \cdot R(L_1) \subset R(L_2) \quad (ii)$$

$$L_1 \cap L_2 = (L_2 \cup L_3) \cap \{d\} \quad (iii)$$

$$L_1 = (R(L_3) \cap L_1) \quad (iv)$$

$$\bar{L}_3 \subset L_1 \quad (v)$$

קבע לכל אחת מהן אם היא נכונה או לא, ונמק את קביעתך.

16. בנה מכונת טיורינג שעל סרט הזיכרון שלה תהיה מילת קלט לא ריקה מעל

הא"ב  $\{a, b\}$ .

המכונה תכתוב את האות האחרונה במלה במקום האות הראשונה. כל האותיות, חוץ מהאות הראשונה, לא ישתנו.

דוגמה

סרט הזיכרון של המכונה לפני תחילת פעולתה:

⊢	a	a	b	a	b	Δ	Δ	.....
---	---	---	---	---	---	---	---	-------

סרט הזיכרון של המכונה בסוף פעולתה:

⊢	b	a	b	a	b	Δ	Δ	.....
---	---	---	---	---	---	---	---	-------

**חישוב מקבילי ומבוזר**

אם למדת מסלול זה, ענה על שתיים מהשאלות 17-20 (לכל שאלה – 25 נקודות).

17. לפניך תכנית, ובה שני תהליכים P1 ו-P2 המתבצעים במקביל. התכנית משתמשת במשתנה גלובלי X מטיפוס שלם, המאותחל ל-1.

```

process P1;
begin
  while X < 0 do
  begin
    X := X + 3;
  end;
  write(X);
end;
    
```

```

process P2;
begin
  while X > 0 do
  begin
    X := X - 3;
  end;
  write(X);
end;
    
```

הערה: בכל אחד משני הסעיפים א-ב אתה נדרש לתאר תסריט. עליך לרשום את התהליך המתבצע בכל שלב, את ההוראה המתבצעת, את ערכי המשתנים, ואת הפלט (אם יש פלט).

- א. תאר תסריט שבו התכנית אינה מסתיימת.
- ב. תאר תסריט סופי שבו מתקבל הפלט: -2, -2.



18. לפניך אלגוריתם, ובו שלושה תהליכים א-ג המתבצעים במקביל. האלגוריתם קולט מספרים שלמים בתחום 1-10 (כולל). בסוף רשימת הקלט יופיע המספר 0 פעמיים. האלגוריתם ידפיס 'כן', אם ברשימת הקלט הופיעו כל המספרים השלמים שבין 1 ל-10 (כולל), אחרת – האלגוריתם ידפיס 'לא'. האלגוריתם משתמש במשתנים הגלובליים:  $P, SUM, N$ , ובמערך  $A$  בגודל 10 שכל איבריו מאותחלים ל-0. כל המשתנים הם מטיפוס שלם.

תהליך ג	תהליך ב	תהליך א
<p>קרא <math>N</math>                      כל עוד <math>N &lt; &gt; 0</math> בצע  <math>A[N] \leftarrow 1</math>                      קרא <math>N</math></p>	<p>קרא <math>N</math>                      כל עוד <math>N &lt; &gt; 0</math> בצע  <math>A[N] \leftarrow 1</math>                      קרא <math>N</math></p>	<p><math>SUM \leftarrow 0</math>                      עבור <math>P</math> מ-1 עד 10 בצע  <math>SUM \leftarrow SUM + A[P]</math>                      אם <math>SUM = 10</math> אזי                      הדפס 'כן'                      אחרת                      הדפס 'לא'</p>

האלגוריתם לא תמיד משיג את מטרתו.

א. נסח, באופן מילולי ועל ידי תרשים תלויות, את דרישות התיאום שיש להוסיף

לאלגוריתם כדי שתמיד ישיג את מטרתו.

ב. העתק למחברתך את התהליכים א-ג, והוסף להם מנגנון תיאום באמצעות

סמפורים, שיבטיח כי האלגוריתם תמיד ישיג את מטרתו.

19. לפניך תכנית, ובה ארבעה תהליכים P1-P4 המתבצעים במקביל. התכנית משתמשת בלוח ופתקים. כל הפתקים על הלוח הם במבנה של תג זיהוי ומספר שלם. על הלוח יש כמה פתקים עם תגי הזיהוי 'A', 'B', 'C', ושלושה פתקים עם התג 'N'. על הפתקים עם תג הזיהוי 'N' רשומים המספרים האלה: על פתק אחד רשום מספר הפתקים מסוג 'A', על פתק שני רשום מספר הפתקים מסוג 'B', ועל פתק שלישי רשום מספר הפתקים מסוג 'C'. הצמדת פתקים אלה אל הלוח נעשית באמצעות התהליך `init_board`. מטרת התכנית היא להדפיס את סכום כל המספרים הרשומים על הפתקים עם תגי הזיהוי 'A', 'B', 'C'.

<pre>program P;</pre>	
<pre>process P1; var   N, S, I, D : integer; begin   S := 0;   removenote ('N', N);   for I := 1 to N do   begin     removenote ('A', D);     S := S + D;   end;   postnote ('S', S); end;</pre>	<pre>process P2; var   N, S, I, D : integer; begin   S := 0;   removenote ('N', N);   for I := 1 to N do   begin     removenote ('B', D);     S := S + D;   end;   postnote ('S', S); end;</pre>
<pre>process P3; var   N, S, I, D : integer; begin   S := 0;   removenote ('N', N);   for I := 1 to N do   begin     removenote ('C', D);     S := S + D;   end;   postnote ('S', S); end;</pre>	<pre>process P4; var   S, D, I : integer; begin   S := 0;   for I := 1 to 3 do   begin     removenote ('S', D);     S := S + D;   end;   writeln (S); end;</pre>
<pre>begin   init_board;   parbegin     P1, P2, P3, P4;   parend; end.</pre>	

**א.** התכנית אינה משיגה את מטרתה. הסבר מדוע.

**ב.** רוני ויעל הציעו שתי דרכים לשנות את התכנית כדי שהיא תשיג את מטרתה.

הדרך של רוני:

נשנה את מבנה הפתקים עם תגי הזיהוי 'A', 'B', 'C', כך שכולם יהיו עם תג

זיהוי אחיד 'G'.

בנוסף, בתהליכים P1, P2, P3 בכל ההוראות מסוג remove\_note יוחלפו

התגים 'A', 'B', 'C' בתג 'G'.

הדרך של יעל:

במקום שלושת הפתקים עם תג הזיהוי 'N', יהיו שלושה פתקים שונים:

פתק אחד עם תג זיהוי 'X' שיכיל את מספר הפתקים עם תג זיהוי 'A',

פתק שני עם תג זיהוי 'Y' שיכיל את מספר הפתקים עם תג זיהוי 'B',

פתק שלישי עם תג זיהוי 'Z' שיכיל את מספר הפתקים עם תג זיהוי 'C'.

בנוסף, נעשה את השינויים האלה:

– בתהליך P1

במקום ההוראה remove\_note('N', N) תהיה ההוראה remove\_note('X', N)

– בתהליך P2

במקום ההוראה remove\_note('N', N) תהיה ההוראה remove\_note('Y', N)

– בתהליך P3

במקום ההוראה remove\_note('N', N) תהיה ההוראה remove\_note('Z', N)

לכל אחת מהדרכים שהוצעו, קבע אם היא מביאה לכך שהתכנית תשיג את

מטרתה, ונמק את קביעתך.

20. קבוצה של  $N$  גנרלים ערוכה סביב מבצר. הגנרלים צריכים להחליט אם לתקוף את המבצר או לסגת. כל גנרל מודיע פעמיים לכל שאר הגנרלים את דעתו בנוגע להחלטה אם לתקוף או לסגת. כל הודעה לכל גנרל נשלחת באמצעות שליח אחר, וכל שליח מגיע ליעדו. כל גנרל מקבל את כל ההודעות שנשלחו אליו, מוסיף להן את דעתו, וסופר את מספר הודעות בעד תקיפה ואת מספר הודעות בעד נסיגה. הוא קובע את החלטתו הסופית על פי הרוב. במקרה של תיקו ההחלטה היא לסגת. ייתכן שיהיו שליחים "מבולבלים" שיעבירו הודעות שקריות.

לפניך האלגוריתם שכל גנרל מבצע:

בחר\_דעה  $X \leftarrow$

שלח לכל שאר הגנרלים  $(X)$

שלח לכל שאר הגנרלים  $(X)$

$A \leftarrow 0$

$R \leftarrow 0$

אם ("התקפה"  $X =$ ) אז  $A \leftarrow 1$

אחרת  $R \leftarrow 1$

עבור  $I$  מ-1 עד  $2 * (N-1)$  בצע

קבל  $M$

אם ("התקפה"  $M =$ ) אז  $A \leftarrow A + 1$

אחרת  $R \leftarrow R + 1$

אם  $A > R$  אז החלט לתקוף

אחרת החלט לסגת

א. הוכח כי לקבוצה של 4 גנרלים, גם בלי הודעות שקריות, ייתכן מצב שבו שני גנרלים יקבעו החלטות שונות.

ב. שנה את האלגוריתם כך שכל הגנרלים יקבעו החלטות זהות כאשר אין הודעות שקריות.

ג. התבסס על האלגוריתם אחרי השינוי שערכת בסעיף ב, והוכח או הפרך את הטענה: בקבוצה של 4 גנרלים מספיק ששליח אחד בלבד יעביר הודעה שקרית, כדי ששני גנרלים יקבעו החלטות שונות.

**שים לב:**

**השאלות במסלול**

**תכנות מונחה עצמים**

**בעמוד הבא.**

**תכנות מונחה עצמים**

אם למדת מסלול זה, ענה על שתיים מהשאלות 21-24 (לכל שאלה – 25 נקודות).

21. עיין בתכנית שלפניך, וענה על הסעיפים א-ד שאחריה (בעמוד הבא).

```
#include <iostream.h>
class A{
public:
    void f ()
    {
        cout << "inside A" << endl;
    }
    void m ()
    {
        cout << "hello" << endl;
    }
private:
    int num;
};
class B : public A{
public:
    void f ()
    {
        cout << "inside B" << endl;
    }
private:
    int num;
};
class C{
public:
    void func (A a)
    {
        a.f();
    }
private:
    int num;
    B b;
};
void main ()
{
    B b;
    C c;
    c.func(b);
}
```

- א.** (i) מהו היחס בין המחלקות A ו- B ? הסבר.  
(ii) מהו היחס בין המחלקות B ו- C ? הסבר.
- ב.** (i) האם אפשר לשנות את הערך של num שבמחלקה A מתוך המחלקה B ?  
אם כן – כתוב כיצד, אם לא – נמק מדוע.  
(ii) האם אפשר לזמן את הפונקציה f() שבמחלקה A מתוך המחלקה B ?  
אם כן – כתוב כיצד, אם לא – נמק מדוע.
- ג.** (i) האם אפשר לזמן את הפונקציה f() שבמחלקה B מתוך המחלקה C ?  
אם כן – כתוב כיצד, אם לא – נמק מדוע.  
(ii) האם אפשר לשנות את הערך של num שבמחלקה C מתוך המחלקה B ?  
אם כן – כתוב כיצד, אם לא – נמק מדוע.  
(iii) האם אפשר לזמן את הפונקציה f() שבמחלקה A מתוך המחלקה C ?  
אם כן – כתוב כיצד, אם לא – נמק מדוע.
- ד.** (i) ניתן לזמן את הפונקציה m() מפונקציית ה- main באמצעות ההוראה:  
b.m();  
הסבר מדוע.  
(ii) האם זימון הפונקציה func מתוך ה- main הוא תקין? אם כן – נמק מדוע,  
וכתוב מה יודפס. אם לא – נמק מדוע.

22. עקוב בעזרת טבלת מעקב אחר התכנית שלפניך, ורשום מה יהיה פלט התכנית.

```
#include <iostream.h>

class A {
public:
    A (int a , int b) { x = a;
                      y = b;
                      cout << "A constructor :'"
                          << '(' << x << ',' << y << ')' << endl; }

    ~ A() { cout << "A destructor :'" << '(' << x << ',' << y << ')' << endl; }

protected:
    int x,y;

};

class B : public A {
public:
    B (double k , int a , int b) : A (a , b)
        { m = k;
          cout << "B constructor :'"
              << '(' << m << ',' << x << ',' << y << ')' << endl; }

    B (double k , int a) : A (a , a)
        { m = k;
          cout << "B constructor :'"
              << '(' << m << ',' << x << ',' << y << ')' << endl; }

    ~ B() { cout << "B destructor :'" << '(' << x << ',' << y << ')' << endl; }

private:
    double m;

};

void main ( )
{
    A s(24 , 6);
    B q(4.5 , 13 , 48);
    B r(3.14 , 20);

}
```

/המשך בעמוד 25/



23. במשרד התקשורת מעוניינים לפתח מערכת תוכנה. המערכת תרכז מידע על מפעילי רשתות טלפונים ניידים, ועל המשתמשים בטלפונים ניידים. יש ארבעה מפעילי רשתות טלפונים ניידים, ולכל אחד מהם יש לא יותר מ-500,000 משתמשים בטלפונים ניידים. כל מפעיל רשת שומר לכל אחד מהמשתמשים בטלפונים ניידים ברשת שלו את הפרטים האלה:
- שם המשתמש, מספר תעודת הזהות שלו, כתובתו, מספר הטלפון של המכשיר הנייד, סוג המכשיר.
- במערכת התוכנה רוצים להגדיר פעולה ClientsPerDevice שתקבל סוג מכשיר, ותדפיס פרטים של כל המשתמשים בסוג המכשיר הזה אצל ארבעת מפעילי הרשתות. הפרטים שיודפסו עבור כל משתמש הם:
- מספר הטלפון של המכשיר הנייד, שם המשתמש, כתובתו, שם מפעיל הרשת.
- א. הגדר בלי מימוש ב- C++ את המחלקות הדרושות לפיתוח מערכת התוכנה. לכל מחלקה ציין תכונות ורק את הפונקציות החברות הדרושות למימוש הפעולה ClientsPerDevice. לכל פונקציה חברה כתוב כותרת ותיעוד.
- ב. הגדר וממש ב- C++ את הפעולה ClientsPerDevice. אתה יכול להיעזר בפעולות שהגדרת בסעיף א בלי לממש אותן.

24. חברה העוסקת בפיתוח משחקי מחשב מעוניינת למחשב את משחק הקלפים "קלף לי", שבו יכולים להשתתף 2-8 שחקנים. במשחק משתתפים "שחקנים אנושיים" (בני אדם) ו"שחקנים ממוחשבים". בכל משחק משתתפים לפחות "שחקן אנושי" אחד ולפחות "שחקן ממוחשב" אחד.

במשחק יש 120 קלפים, ולכל קלף יש שלושה מאפיינים:

– ערך – מספר שלם בתחום 1-12;

– צורה מבין שלוש הצורות – עיגול, ריבוע, משולש;

– צבע מבין ארבעת הצבעים – אדום, צהוב, כחול, ירוק.

במשחק יש שתי ערמות של קלפים: ערמה אחת היא קופה משותפת וסדר הקלפים בה אקראי, וערמה שנייה שמתחילה בקלף גלוי אחד ונערמת לפי כללי המשחק. בתחילת המשחק כל שחקן מקבל 5 קלפים מהקופה. קלף נוסף מהקופה נחשף ומונח בנפרד גלוי לעיני כל השחקנים, ועליו תיערם הערמה השנייה. כל שחקן רשאי להניח בתורו קלף אחד על גבי הקלף החשוף, על פי ארבעת הכללים האלה:

\* שחקן יכול להניח קלף, אם הוא מאותה צורה או מאותו צבע של הקלף הנמצא

בראש הערמה, או קלף שהערך שלו גבוה מן הערך של הקלף שבראש הערמה.

\* אם אין לשחקן קלף שהוא יכול להניח בראש הערמה, עליו למשוך קלף אחד מן

הקופה, והתור עובר לשחקן הבא.

\* המנצח במשחק הוא השחקן הראשון שנותר בלי קלפים.

\* אם הקופה התרוקנה מקלפים לפני שיש מנצח, ואין שחקן שיכול להניח קלף נוסף,

המשחק יסתיים בלי הכרעה.

הקלפים של ה"שחקן הממוחשב" ממוינים תמיד לפי ערך הקלף בסדר עולה.

**(שים לב: המשך השאלה בעמוד הבא.)**

לפניך חלק מההגדרות בלי מימוש של שלוש המחלקות:

קלף (Card), ערמת קלפים (Collection), שחקן (Player).

```
class Card
{
    public:
        int getValue ();           // אחזור ערך של קלף
        int getShape ();          // אחזור צורה של קלף
        int getColor ();          // אחזור צבע של קלף
    private:
        int value;                // ערך של קלף
        int shape;                // צורה של קלף
        int color;                // צבע של קלף
};

class Collection
{
    public:
        void addCard (Card x);    // הוספת קלף לערמה
        Card removeCard ();       // הוצאת קלף מערמה והחזרתו
        int getNumOfCards ();     // החזרת מספר הקלפים בערמה
    private:
        int numOfCards;          // מספר קלפים בערמה
        Card cards[120];         // הקלפים בערמה
};

class Player
{
    public:
        void moveCard(collection x); // הוצאת קלף מידי השחקן והנחתו בראש ערמה
    private:
        int playerNum;           // מספר שחקן
        Collection hand;         // הקלפים שבידי השחקן
};
```

- א. הגדר בלי מימוש ב- C++ את המחלקות "שחקן אנושי" ו"שחקן ממוחשב".  
ל"שחקן הממוחשב" בשונה מה"שחקן האנושי" יש פעולה מיוחדת של מיון הקלפים.
- ב. ממש במחלקה Player פעולה שתמשוך קלף מן הקופה ותעביר אותו לשחקן.
- ג. הגדר ב- C++ את המשתנים הדרושים לניהול המשחק "קלף ליי".

## ב ה צ ל ח ה !

זכות היוצרים שמורה למדינת ישראל  
אין להעתיק או לפרסם אלא ברשות משרד החינוך התרבות והספורט