

## 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, —  $(25 \times 2)$  — 50 נקודות  
ומהן יש לענות על שתיים.
- פרק שני – בפרק זה שאלות בחמישה מסלולים שונים. —  $(25 \times 2)$  — 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 אֲנִי C.
  - רשום על הכריכה החיצונית של המחברת באיזו שפה אתה כותב – פסקל אֲנִי 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.

כתוב במחברת הבחינה בלבד, בעמודים נפרדים, כל מה שברצונך לכתוב כטייטה (ראשי פרקים, חישובים וכדומה).  
רשום "טייטה" בראש כל עמוד טייטה. רישום טייטות 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 1-4 (לכל שאלה – 25 נקודות).

1. לפניך קטע תכנית בפסקל וב-C:

#### פסקל

```

const N = 8;
type typeArray = array [1.. N] of integer;

function firstCheck (x , num : integer) : boolean;
    { הפונקציה מקבלת ספרה x ומספר שלם num גדול מאפס.
    { הפונקציה מחזירה...
begin
    if num = 0
        then firstCheck := false
    else if (num mod 10 = x)
        then firstCheck := true
    else firstCheck := firstCheck (x , num div 10);
end;

function secondCheck (a : typeArray ; k : integer) : integer;
    { הפונקציה מקבלת מערך a בגודל N, המכיל מספרים שלמים גדולים מאפס
    { ומספר שלם k בתחום 1 עד N (כולל).
    { הפונקציה מחזירה...
begin
    if (k = N)
        then secondCheck := 0
    else if (firstCheck (a[k] mod 10 , a[k+1]))
        then secondCheck := 1 + secondCheck (a , k + 1)
    else secondCheck := secondCheck (a , k + 1);
end;
/המשך בעמוד 3/

```

C

```
typedef N = 8;
int firstCheck (int x , int num)
// הפונקציה מקבלת ספרה x ומספר שלם num גדול מאפס.
// הפונקציה מחזירה ...
{
    if (num == 0)
        return 0;
    else
        if (num % 10 == x)
            return 1;
        else
            return firstCheck(x , num /10);
};
```

```
int secondCheck (int a[] , int k)
// הפונקציה מקבלת מערך a בגודל N, המכיל מספרים שלמים גדולים מאפס
// ומספר שלם k בתחום 0 עד N-1 (כולל).
// הפונקציה מחזירה ...
{
    if (k == (N-1))
        return 0;
    else
        if (firstCheck (a[k] % 10 , a[k+1]) == 1)
            return 1+secondCheck (a , k+1);
        else
            return secondCheck (a , k+1);
};
```

א. מה מבצעת הפונקציה firstCheck ?

ב. נתון מערך חד-ממדי a בגודל 8.

→ תחילת המערך

400	200	100	25	351	33	34	4
-----	-----	-----	----	-----	----	----	---

מה יחזיר הזימון secondCheck (a , 4) ?

רשום את המעקב אחר ביצוע הפונקציה secondCheck. אין צורך להראות מעקב

אחר הפונקציה firstCheck.

ג. מה מבצעת הפונקציה secondCheck ?

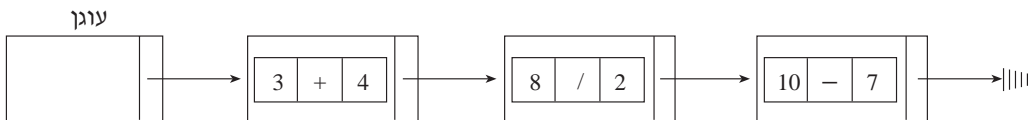
/המשך בעמוד 4/

2. רשימה "חשבונית" L היא רשימה ששדה התוכן שלה מייצג ביטוי חשבוני.

הביטוי החשבוני מורכב משלושה חלקים:

- מספר שלם גדול מאפס
- תו אחד מבין ארבעת התווים:
  - + המייצג חיבור
  - המייצג חיסור
  - \* המייצג כפל
  - / המייצג חילוק
- מספר שלם גדול מאפס

דוגמה לרשימה "חשבונית" L:



**א.** הגדר בסביבת העבודה את טיפוס שדה התוכן של איבר ברשימה "חשבונית" L.

**ב.** ממש בסביבת העבודה תת-תכנית calculate, שתקבל רשימה "חשבונית" L ומקום p ברשימה. p הוא מקום ב-L שאינו סוף-רשימה ואינו עוגן-רשימה. התת-תכנית תחזיר את התוצאה המתקבלת מהביטוי החשבוני הנמצא באיבר שבמקום p.

**ג.** ממש בסביבת העבודה תת-תכנית sumExpressions, שתקבל רשימה "חשבונית" L, ותחזיר את הסכום הכולל של תוצאות הביטויים החשבוניים הנמצאים ברשימה. בעבור רשימה ריקה יוחזר 0. עליך להשתמש בתת-תכנית calculate.

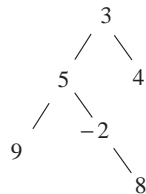
בעבור הרשימה "החשבונית" L בדוגמה הנתונה, התת-תכנית sumExpressions תחזיר 14.

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

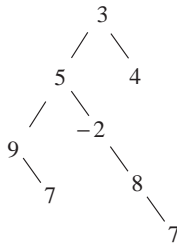
3. לפניך פעולה:

<p>הפעולה מקבלת מספר שלם <math>N</math> ועץ בינארי <math>T</math> לא ריק שערכיו הם מספרים שלמים. לכל עלה בעץ <math>T</math> שערכו גדול מ-<math>N</math>, הפעולה מוסיפה בן ימני שערכו <math>N</math>. הנחה: העץ <math>T</math> מאותחל.</p>	<p><b>הוסף_עלים (T, N)</b></p>
---	--------------------------------

דוגמה: נתון העץ  $T$



לאחר זימון הפעולה **הוסף\_עלים (T, 7)**, העץ  $T$  הוא:



כתוב אלגוריתם שיממש את הפעולה **הוסף\_עלים (T, N)**.

4. במכון "ארוביקה" יש שיעורי התעמלות אֶרֶוֹבִיט. השיעורים מתקיימים בימים א-ה בשעות

7:00-22:00. כל שיעור מתחיל בשעה עגולה ונמשך שעה אחת. בכל שעה פועל במכון שיעור

אחד בלבד, ויכולים להשתתף בו לכל היותר 25 אנשים.

אדם יכול להירשם לשיעור אחד בלבד. בהרשמה הוא מציין את היום בשבוע ואת השעה

של השיעור שאליו הוא מבקש להירשם, ומוסר את שמו ואת מספר הטלפון הנייד שלו.

ההרשמה לשיעור, ביום ובשעה המבוקשים, היא למשך שנה שלמה.

אם אדם מעוניין להירשם לשיעור מסוים ואין מקום בשיעור, הוא נרשם כממתין ומוסר

את הפרטים האלה: השם, מספר הטלפון הנייד, היום והשעה של השיעור המבוקש.

**א.** יצג את המידע הדרוש לניהול הרישום לשיעור במכון "ארוביקה".

**ב.** כתוב אלגוריתם **משתתפים\_בשיעור** שידפיס את השמות ואת מספרי הטלפון של

כל המשתתפים בשיעור מסוים. רשום את הפרמטרים שהאלגוריתם מקבל ותיעוד.

**ג.** כתוב אלגוריתם **מקום\_פנוי** שיקבל יום ושעה של שיעור שיש בו מקום פנוי, וישבץ

אליו את הראשון מבין הממתנים שזה השיעור שביקש. האלגוריתם ימחק את

רישומו של האדם כממתין. רשום את הפרמטרים שהאלגוריתם מקבל ותיעוד.

/המשך בעמוד 6/

**פרק שני (50 נקודות)**

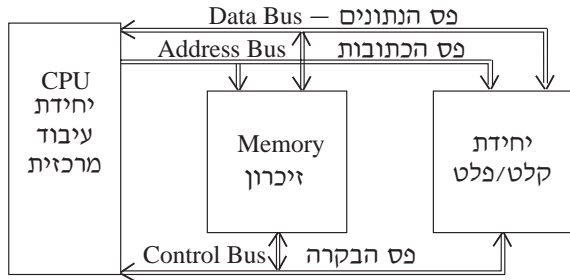
בפרק זה שאלות בחמישה מסלולים שונים. ענה רק על שאלות במסלול שלמדת.

**מערכות מחשב ואסמבלר**

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

5. לפניך תיאור סכמתי של מערכת ובה יחידת עיבוד מרכזית, פסי מעבד, זיכרון,

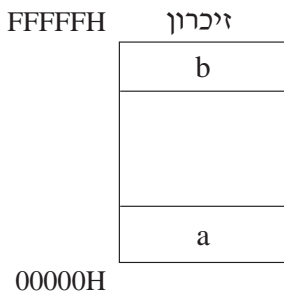
יחידת קלט / פלט.



הזיכרון מכיל יחידת זיכרון לקריאה/כתיבה ויחידת זיכרון לקריאה בלבד. אחת מהן ממוקמת בתחילת מרחב הכתובות של הזיכרון, והאחרת ממוקמת בסוף מרחב הכתובות של הזיכרון. הגודל של כל יחידה הוא 64K בתים.

א. לפניך איור של הזיכרון ובו מסומנים האזורים: a המתחיל בכתובת 00000H,

ו-b המסתיים בכתובת FFFFFFFH.



(1) איזו יחידת זיכרון יש למקם בתחילת מרחב

הכתובות של הזיכרון – אזור a,

ואיזו יחידת זיכרון יש למקם בסוף מרחב

הכתובות של הזיכרון – אזור b? נמק.

(2) מהי כתובת הסיום של יחידת הזיכרון הממוקמת

באזור a, ומהי כתובת ההתחלה של יחידת הזיכרון

הממוקמת באזור b?

ב. מידע עובר מיחידת העיבוד המרכזית בפס הכתובות ובפס הנתונים אל היחידות

השונות.

באילו קווים בפס הבקרה משתמשת יחידת העיבוד המרכזית כדי לקבוע איזו

יחידה תקבל את המידע? נמק. בתשובתך התייחס לכל אחת משלוש היחידות:

יחידת קלט/פלט, יחידת זיכרון לקריאה/כתיבה, יחידת זיכרון לקריאה בלבד.

6. בתחילת תכנית בשפת אסמבלר הוגדרו נתונים בצורה זו:

```
DATA SEGMENT
    KAM DB 04H
    ARR1 DB 04H , 10H , 1CH , 5DH
    ARR2 DB 02H , 08H , 0DH , 1EH
DATA ENDS
```

שים לב: הנתונים והכתובות הם הקסדצימליים.

לפניך קטע התכנית:

```
MOV     AX , DATA
MOV     DS , AX
XOR     CX , CX
LEA     BX , ARR1
LEA     DI , ARR2
LEA     SI , KAM
MOV     CL , KAM
MOV     AH , 0
AGAI:   MOV     DH , [DI]
        INC     DI
        SHL     DH , 1
        CMP     DH , [BX]
        JNZ     COUNT1
        INC     AH
COUNT1: MOV     [BX] , DH
        INC     BX
        LOOP    AGAI
        MOV     [SI] , AH
SOF:    NOP
```

א. עקוב בעזרת טבלת מעקב אחר ביצוע קטע התכנית, ורשום מה יהיה התוכן של

תאי הזיכרון.

ב. מה יהיה התוכן של תא הזיכרון KAM בסיום קטע התכנית?

ג. מה מבצע קטע התכנית?

7. סדרה חשבונית היא סדרה שבה ההפרש בין שני איברים עוקבים הוא גודל קבוע. תלמיד כתב בשפת אסמבלר קטע תכנית לחישוב האיברים של סדרה חשבונית. האיבר הראשון של הסדרה נמצא בתא הזיכרון FIRST, וההפרש בין שני איברים עוקבים נמצא בתא הזיכרון DIFF. קטע התכנית סופר כמה איברים אי-זוגיים יש בסדרה. חישוב הסדרה יסתיים כאשר חושבו COUNT איברים או כאשר יש גלישה באיבר האחרון שחושב. הנתונים הוגדרו בצורה זו:

DATA SEGMENT

```
FIRST          DB 243
DIFF           DB 7
COUNT         DW 5
DATA ENDS
```

לפניך קטע התכנית שכתב התלמיד. מהקטע הושמטו פקודות ואופרנדים.

```
START:  MOV     AX, DATA
        MOV     DS, AX
        MOV     CX, COUNT
        MOV     DL, 0
        MOV     BL, DIFF
        XOR     AX, AX
        MOV     AL, FIRST
AGAIN:  AND     AL, [ ]
        JZ      NEXT
        INC     DL
NEXT:   MOV     AL, [ ]
        ADD     AL, BL
        [ ]     FINISH
        MOV     FIRST, AL
        [ ]     [ ]
FINISH: NOP
```

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



8.

במפעל למשקאות ערכו מבחן טעימה ל-8 סוגים חדשים של מיצים שסומנו במספרים 0-7. במבחן הטעימה השתתפו 5 טועמים מומחים. כל טועם בדק כל סוג של מיץ פעם אחת, וסימן בטופס מיוחד אם סוג המיץ לטעמו (1) או לא (0).

דוגמה: לפניך טופס של אחד הטועמים שציין שסוגי המיצים 0, 4, 5, 7 הם לטעמו:

סוג המיץ	0	1	2	3	4	5	6	7
תוצאות הטעימה	1	0	0	0	1	1	0	1

תוצאות הטעימה של הטועמים מאוחסנות בזיכרון החל מהכתובת TASTER, כל טופס מאוחסן בבית אחד.

עבור כל סוג מיץ נצבר במערך RESULTS מספר הטועמים שמיץ זה היה לטעמם.

הנתונים הוגדרו כך:

תוצאות הטעימה של הטועמים: TASTER DB 0B1H, 03H, 2BH, 0A6H, 0AAH

מערך לצבירת תוצאות הטעימה: RESULTS DB 8 DUP(0)

שים לב: הנתונים הם הקסדצימליים.

לפניך קטע תכנית המטפל בתוצאות טעימה של טועם אחד:

```

                                LEA      BX, TASTER
                                MOV      DL, 1
                                MOV      SI, 0
                                MOV      AL, [BX]
LOOP1:                          SHR      AL, 1
                                JC       YES
                                CMP      SI, 7
                                JZ      FINISH
                                INC      SI
                                JMP      LOOP1
                                YES:     ADD      RESULTS[SI], DL
                                JMP      LOOP1
                                FINISH:
    
```

**א.** קטע התכנית אינו מבצע את הנדרש. הוסף הוראות, כדי שקטע התכנית יבצע את הנדרש. ציין היכן בקטע התכנית תמקם את ההוראות שהוספת.

**ב.** הוסף הוראות לקטע התכנית, כדי שיטפל בתוצאות הטעימה של 5 טועמים. הקטע יסכם לכל סוג מיץ את מספר הטועמים שציינו שהוא לטעמם, ויאחסן תוצאה זו במקום המתאים במערך RESULTS.

הערה: בסעיפים א-ב אין צורך להעתיק את קטע התכנית.

**תורת המחשב**

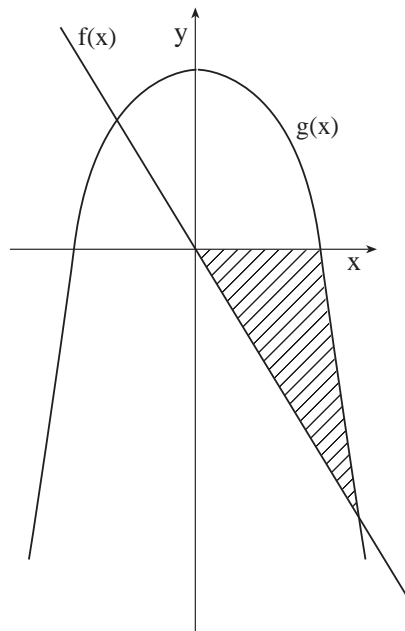
אם למדת מסלול זה, ענה על שתיים מהשאלות 9-12 (לכל שאלה – 25 נקודות).

9. נתונות שתי הפונקציות:

$$f(x) = -2x$$

$$g(x) = -x^2 + 6$$

כתוב תכנית, שתשתמש בשיטת הטרפז, לחישוב השטח התחום בין הפונקציה  $f(x)$ , הפונקציה  $g(x)$  וציר ה- $x$  (השטח המקווקו בציר), בדיוק של  $10^{-6}$ .



10. נגדיר סדרה של גרפים לא מכוונים באופן הזה:

בגרף  $G_0$  יש צומת אחד, המסומן ב-  $V_0$ , ואין בו קשתות.

כל גרף  $G_n$  יוגדר בעזרת הגרף  $G_{n-1}$  באופן הזה:

– נוסיף לגרף  $G_{n-1}$  צומת  $V_n$ .

– אם  $n$  זוגי, נוסיף לגרף  $G_{n-1}$  קשתות שיחברו את הצומת  $V_n$

אל כל הצמתים שהאינדקס שלהם אי-זוגי.

– אם  $n$  אי-זוגי, נוסיף לגרף  $G_{n-1}$  קשתות שיחברו את הצומת  $V_n$

אל כל הצמתים שהאינדקס שלהם זוגי.

א. צייר את הגרפים:  $G_0, G_1, G_2, G_3, G_4$ . שים לב: 0 הוא מספר זוגי.

ב. מה תהיה דרגת הצומת  $V_n$  בגרף  $G_n$  ?

ג. מה תהיה דרגת הצומת  $V_n$  בגרף  $G_{n+1}$  ?

ד. לפניך טענה.

עבור  $n$  כלשהו שלם וחיובי:

לכל הצמתים בגרף  $G_n$  יש אותה דרגה

או

קיימים שני מספרים שלמים וחיוביים  $x$  ו-  $y$ , כך שלחלק מהצמתים

בגרף  $G_n$  יש דרגה  $x$  ולשאר הצמתים בגרף  $G_n$  יש דרגה  $y$ .

קבע אם הטענה נכונה או אינה נכונה, ונמק את קביעתך.

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

$$L = \{ a^m b^n a^n \mid m, n \geq 1, (m+n) \text{ זוגי} \}$$

בנה אוטומט מחסנית שיקבל את השפה  $L$ .

12. ללימודים במכללה מסוימת מתקבלים תלמידים על פי ציוניהם בבחינות הבגרות.

הקריטריונים לקבלה ללימודים במכללה הם אלה:

- \* תעודת בגרות ובה לפחות 5 מקצועות.
- \* הציונים במתמטיקה, באנגלית ובמקצוע נוסף צריכים להיות גבוהים מ- 84.
- \* כל הציונים חייבים להיות גבוהים מ- 74.

במכללה מקודדים את הציונים באופן הזה:

A : 100-95

B : 94-85

C : 84-75

D : 74-55

האותיות המייצגות את הציונים מוקלדות כמחרוזת משמאל לימין באופן הזה:

האות הראשונה שמוקלדת מייצגת את הציון במתמטיקה,

האות השנייה שמוקלדת מייצגת את הציון באנגלית,

ולאחר מכן מוקלדות האותיות המייצגות את הציונים במקצועות האחרים.

א. לפיך חמש מחרוזות, v-i.

כל אחת מהמחרוזות מייצגת ציונים של תלמיד המבקש להתקבל ללימודים במכללה.

קבע לכל אחת מהמחרוזות אם התלמיד יתקבל או לא יתקבל ללימודים במכללה.

נמק את קביעותיך.

AACCCBBCA i

ABABDBB ii

BBBBBBA iii

AAAA iv

AACCCC v

ב. בנה אוטומט סופי דטרמיניסטי, שיקבל כקלט מחרוזת המייצגת ציונים של תלמיד

ויבדוק אם התלמיד יתקבל ללימודים במכללה.

**מודלים חישוביים**

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

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

$$L = \{ a^m b^n a^n \mid m, n \geq 1, \text{ זוגי } (m+n) \}$$

בנה אוטומט מחסנית שיקבל את השפה  $L$ .

14. בנה מכונת טיורינג שתקבל כקלט מילה מעל הא"ב  $\{a, b, c\}$ .

המכונה תבדוק אם לפני כל  $a$ , בצמוד לו, יש  $c$  אחד לפחות.

אם כן – המכונה תעצור במצב מקבל, אם לא – המכונה לא תעצור.

אם הקלט הוא המילה הריקה, המכונה תעצור במצב מקבל.

דוגמה:

בעבור מילת הקלט  $bccabcbeccab$  המכונה תעצור במצב מקבל.

בעבור מילת הקלט  $cbabbcab$  המכונה לא תעצור.

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

$$L_1 = \{ a^n b^{n+2} \mid n \geq 0 \}$$

$$L_2 = \{ a^n b^m \mid n, m \geq 0, \text{ זוגי } n \}$$

$$L_3 = \{ a^{2n} b^{n+3} \mid n \geq 0 \}$$

א. קבע לכל אחת מהשפות  $L_1, L_2, L_3$  אם היא רגולרית או אינה רגולרית.

נמק את קביעותיך.

ב. מהי השפה  $L_1 \cap L_2$ ? נמק.

ג. האם מספר ה- $b$ ים במילה השייכת לשפה  $L_2 \cdot R(L_2)$  הוא זוגי? נמק.

ד. קבע לכל אחת מהמילים  $w_1, w_2$ , אם היא שייכת לשפה  $\bar{L}_1 \cap L_3$ .

נמק את קביעותיך.

$$w_1 = aaaabbbbb$$

$$w_2 = bbb$$

16. ללימודים במכללה מסוימת מתקבלים תלמידים על פי ציוניהם בבחינות הבגרות. הקריטריונים לקבלה ללימודים במכללה הם אלה:
- \* תעודת בגרות ובה לפחות 5 מקצועות.
  - \* הציונים במתמטיקה, באנגלית ובמקצוע נוסף צריכים להיות גבוהים מ- 84.
  - \* כל הציונים חייבים להיות גבוהים מ- 74.

במכללה מקודדים את הציונים באופן הזה:

A : 100-95

B : 94-85

C : 84-75

D : 74-55

האותיות המייצגות את הציונים מוקלדות כמחרוזת משמאל לימין באופן הזה: האות הראשונה שמוקלדת מייצגת את הציון במתמטיקה, האות השנייה שמוקלדת מייצגת את הציון באנגלית, ולאחר מכן מוקלדות האותיות המייצגות את הציונים במקצועות האחרים.

א. לפניך חמש מחרוזות, v-i.

כל אחת מהמחרוזות מייצגת ציונים של תלמיד המבקש להתקבל ללימודים במכללה.

קבע לכל אחת מהמחרוזות אם התלמיד יתקבל או לא יתקבל ללימודים במכללה.

נמק את קביעותיך.

AACCCBBCA i

ABABDBB ii

BBBBBBA iii

AAAA iv

AACCCC v

ב. בנה אוטומט סופי דטרמיניסטי, שיקבל כקלט מחרוזת המייצגת ציונים של תלמיד ויבדוק אם התלמיד יתקבל ללימודים במכללה.

**חישוב מקבילי ומבוזר**

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

17. לפניך תכנית ובה שני תהליכים Proc1 ו- Proc2 המתבצעים במקביל. התכנית משתמשת במשתנה גלובלי N.

<pre>program TestSh; var N: integer;</pre>	
<pre>process Proc1; var A: integer; begin   read(A);   while N &lt; A do   begin     N := N + 1;     write(N);   end; end;</pre>	<pre>process Proc2; var B, C: integer; begin   read(B);   C := B;   while B &gt; 0 do   begin     B := B - 2;     if B = N then     begin       N := 0;       B := C;     end;   end; end;</pre>
<pre>begin {main}   N := 0;   parbegin     Proc1;     Proc2;   parend; end.</pre>	

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

**א.** בעבור הקלט 3 ל-A ו-4 ל-B הצג תסריט עד שיתקבלו הפלטים (משמאל לימין):  
 . 1 , 2 , 1 , 2

**ב.** בעבור הקלט 1 ל-A ו-3 ל-B הצג תסריט עד שיתקבלו הפלטים (משמאל לימין):  
 . 0 , 0 , 1

18.

במצפה הכוכבים "עין לשחקים" נותנים שירותי טלסקופ דרך האינטרנט למנויים בלבד. לכל מנוי יש מספר מנוי. אין שני מנויים עם אותו מספר מנוי. התור לשימוש בטלסקופ מנוהל כמערכת מבוזרת בעזרת תוכנה המותקנת על מחשבי המנויים. מחשבי המנויים מחוברים תמיד לרשת, ואינם מתנתקים אף פעם. התוכנה בנויה על בסיס אלגוריתם RA, ופועלת כל הזמן במחשב של כל מנוי. בתוכנה יש שני תהליכים: תהליך ראשי ותהליך לטיפול בבקשות, הפועלים במקביל ומאפשרים ניהול תור בלי מחשב מרכזי. כאשר שני מנויים או יותר מבקשים להשתמש בטלסקופ באותו זמן, ניתנת עדיפות למנוי שמספר המנוי שלו קטן יותר.

לפניך האלגוריתם:

תהליך ראשי:
<p>"אני רוצה להשתמש בטלסקופ" → false</p> <p><b>בצע תמיד:</b></p> <p>קטע לא קריטי { מהלך חיים רגיל }</p> <p>"אני רוצה להשתמש בטלסקופ" → true</p> <p><b>עבור כל מחשב Z שאינו המחשב שלי בצע:</b></p> <p><b>שלח בקשה</b> (כתובת מחשב Z, מספר המנוי שלי, כתובת המחשב שלי)</p> <p><b>עבור מספר מנויים פחות אחד בצע:</b></p> <p><b>קבל אישור</b></p> <p>{ השתמש בטלסקופ }</p> <p>"אני רוצה להשתמש בטלסקופ" → false</p> <p><b>עבור כל המנויים הממתינים בצע:</b></p> <p><b>שלח אישור</b></p>
תהליך לטיפול בבקשות:
<p><b>בצע תמיד:</b></p> <p><b>קבל בקשה</b> (מספר המנוי של השולח, כתובת מחשב שולח)</p> <p><b>אם לא</b> "אני רוצה להשתמש בטלסקופ"</p> <p><b>שלח אישור</b> (כתובת מחשב שולח)</p> <p><b>אחרת אם</b> מספר המנוי שלי &lt; מספר המנוי של השולח <b>אזי</b></p> <p><b>שלח אישור</b> (כתובת מחשב שולח)</p> <p><b>אחרת</b></p> <p><b>הכנס לרשימת הממתינים</b> (כתובת מחשב שולח)</p>



בסעיפים א-ג שלפניך, כאשר אתה מציג תסריט עליך לרשום את התהליך המתבצע, את ההוראה המתבצעת, ואת ערכי הפרמטרים בהוראות הכוללות פרמטרים.

**א.** נניח שמנוי 100 אינו מעוניין להשתמש בטלסקופ, והוא מקבל ממנוי 50 וממנוי 102 בקשות להשתמש בטלסקופ.

i האם מנוי 100 שולח אישור לשניהם? נמק את תשובתך.

ii האם מנוי 100 יכול להניח שמנוי 50 ישתמש בטלסקופ לפני מנוי 102? נמק את תשובתך.

**ב.** הנח ששני המנויים היחידים במערכת הם מנוי 1 ומנוי 2.

שניהם שולחים זה לזה בקשות לשימוש בטלסקופ.

בהנחה שכל אחד משני מנויים אלה שולח בקשה למנוי האחר לפני שהוא מקבל את

הבקשה שנשלחה אליו, האם ייתכן שמנוי 1 יוכל להשתמש בטלסקופ בלי שמנוי 2

יימצא ברשימת הממתינים שלו?

אם כן – הצג תסריט מתאים, אם לא – הסבר מדוע.

**ג.** הצג תסריט שבו שני מנויים פונים יחד לשימוש בטלסקופ (נמצאים יחד בקטע

הקריטי).

19.

נבחרת בית ספר בקפיצה לגובה, שבה 12 קופצים, מקיימת תחרות הכנה פנימית. כל קופץ מקבל מספר בין 1 ל-12, והוא קופץ שלוש קפיצות. התוצאה הקובעת בעבור כל קופץ היא התוצאה הטובה ביותר מבין התוצאות של שלוש הקפיצות שלו. כדי למצוא את תוצאת הקפיצה הטובה ביותר מבין כל התוצאות הקובעות של 12 הקופצים בנבחרת, פיתחו אלגוריתם הכולל 4 תהליכים המתבצעים במקביל:

- \* תהליך 1, תהליך 2, תהליך 3 – הם תהליכים זהים.
- כל תהליך קולט שלוש תוצאות של קופץ, ומוצא את התוצאה הקובעת בעבורו.
- \* תהליך 4 מוצא את התוצאה הטובה ביותר מבין כל התוצאות הקובעות של 12 הקופצים.

האלגוריתם משתמש במשתנים הגלובליים מטיפוס שלם: TotalMax, TempMax, N ; ובסמפורים: Sem1, Sem2.

כמו כן, האלגוריתם משתמש בפעולה האטומית קלוט-קופץ (P, A, B, C) הממתינה עד שמתקבלים 4 קלטים:

למשתנה P – מספר הקופץ, ולמשתנים A, B, C – תוצאות הקפיצות של הקופץ P.

<p><b>תהליך 3</b> { זהה לתהליך 1 }</p>	<p><b>תהליך 2</b> { זהה לתהליך 1 }</p>	<p><b>תהליך 1</b>                      כל עוד <math>N \leq 12</math> בצע  <math>N \leftarrow N + 1</math>                      קלוט-קופץ (P, A, B, C)  <math>Max \leftarrow A</math>                      אם <math>B &gt; Max</math> אזי  <math>Max \leftarrow B</math>                      אם <math>C &gt; Max</math> אזי  <math>Max \leftarrow C</math>                      wait (Sem1)  <math>TempMax \leftarrow Max</math>                      signal (Sem2)</p>
<p><b>תהליך 4</b>                      עבור I מ-1 עד 12 בצע                      wait (Sem2)                      אם <math>TempMax &gt; TotalMax</math> אזי  <math>TotalMax \leftarrow TempMax</math>                      signal (Sem1)</p>		
<p><b>האלגוריתם הראשי</b>  <math>N \leftarrow 1</math>  <math>TempMax \leftarrow 0</math>  <math>TotalMax \leftarrow 0</math>  <math>\{ \text{סמפור} \} Sem1 \leftarrow 1</math>  <math>\{ \text{סמפור} \} Sem2 \leftarrow 0</math>                      parbegin                      תהליך 1; תהליך 2; תהליך 3; תהליך 4;                      parend                      הדפס TotalMax</p>		

- א. באיזה מנגנון תיאום האלגוריתם משתמש, וכיצד מנגנון זה בא לידי ביטוי באלגוריתם?
- ב. באלגוריתם הנתון יש כשל מקבילי, שבגללו אחד או יותר מהתהליכים אינו מסתיים, ולכן באלגוריתם הראשי לא מודפס דבר.
- i הסבר מדוע נוצר הכשל המקבילי הזה.
- ii הוסף לאלגוריתם הוראות כדי למנוע את הכשל המקבילי.
- ג. המאמן מעוניין לקבל את מספרו של הקופץ שהשיג את התוצאה הטובה ביותר. הִרְחֵב את האלגוריתם המקבילי כדי שגם מטרה זו תושג.

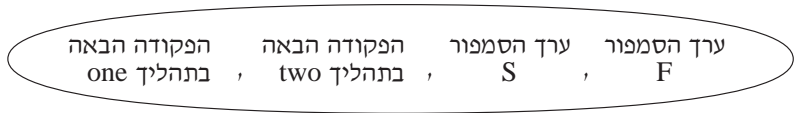
/המשך בעמוד 20/

20. לפניך תכנית מקבילית:

<pre> program best; var F: semaphore := init(1);     S: semaphore := init(1);         </pre>	
<pre> process one; begin     wait(S);     wait(F);     { קטע קריטי }     signal(F); end;         </pre>	<pre> process two; begin     wait(F);     { קטע קריטי }     signal(S); end;         </pre>
<pre> begin     parbegin         one; two;     parend; end.         </pre>	

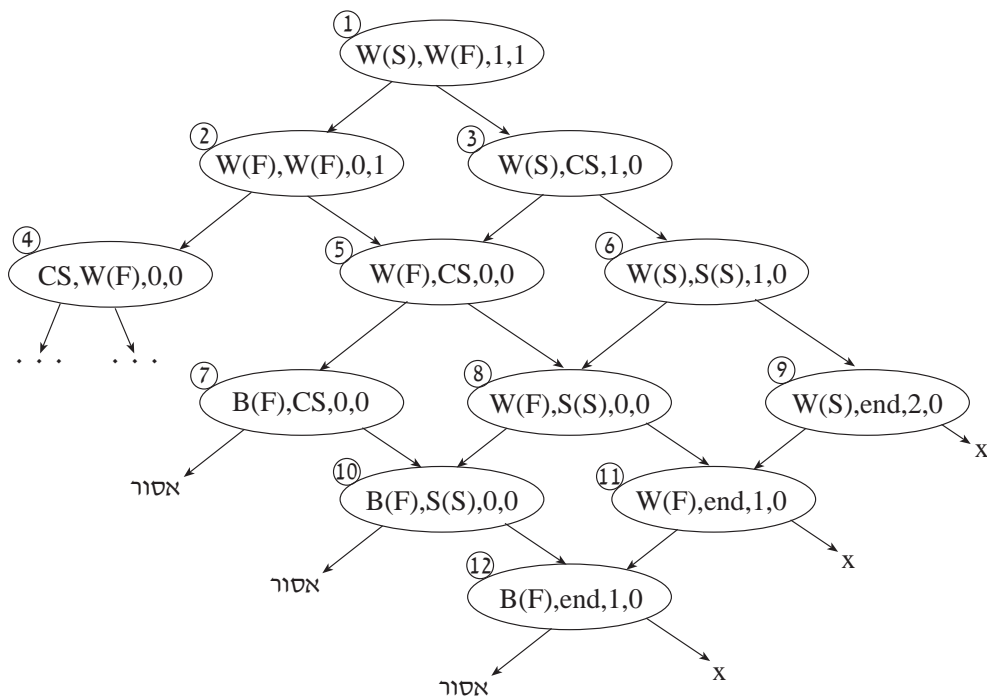
תלמיד בנה תרשים מצבים לתכנית, אך לא סיים אותו.

סדר הצגת המידע במצב בתרשים מתואר בציור שלפניך:



CS מייצג את הקטע הקריטי.

B מייצג תהליך חסום על ידי סמפור.



- א. העתק למחברתך את מצב מספר 4, והשלם את תרשים המצבים. מספר את המצבים שהוספת ממספר 13 ואילך.
- ב. האם ייתכן קיפאון בתכנית? נמק בעזרת תרשים המצבים.
- ג. האם יש בתכנית מניעה הדדית? הוכח בעזרת תרשים המצבים או באופן מילולי.

**תכנות מונחה עצמים**

אם למדת מסלול זה ואתה כותב בג'אווה, ענה על שתיים מהשאלות 21-24 שבעמודים 22-28.  
(לכל שאלה – 25 נקודות)

אם למדת מסלול זה ואתה כותב ב-C#, ענה על שתיים מהשאלות 25-28 שבעמודים 30-36.  
(לכל שאלה – 25 נקודות)

21. לפניך תכנית:

```
public class Stam
{
    public void B(int m , int n)
    {   System.out.println (m  n);   }
    public void f ()
    {   System.out.println("inside Stam");   }
}
public class B extends Stam
{
    protected int y;
    public B(int x)
    {   this.y = x;   }
}
public class C extends Stam
{
    public int x;
    public static void g()
    {   System.out.println("inside C ");   }
}

public class Test
{
    public static void main (String[] args)
    {
        C temp = new C();
    }
}
```

ענה על הסעיפים א-ה, בהתייחס לתכנית הנתונה.

**א.** האם אפשר לשנות את הערך של התכונה  $x$  המוגדרת במחלקה C, מתוך שיטה במחלקה Stam?

אם כן – כתוב כיצד, אם לא – נמק מדוע.

**ב.** i כתוב בשיטה main, הוראה או הוראות לזימון השיטה B המוגדרת במחלקה Stam.

ii כתוב בשיטה main, הוראה או הוראות לזימון השיטה B המוגדרת במחלקה B.

**ג.** לפניך שלוש הוראות, i-iii.

לכל אחת מהן, קבע אם אפשר להוסיף אותה בשיטה main. נמק את קביעותיך בעזרת מאפיינים של תכנות מונחה עצמים.

Stam obj = new B(7); i

Stam obj = new B(); ii

B obj = new Stam(); iii

**ד.** האם אפשר לזמן את השיטה f() המוגדרת במחלקה Stam, מתוך שיטה במחלקה B?

אם כן – כתוב את הזימון, אם לא – נמק מדוע.

**ה.** האם אפשר לזמן את השיטה g() המוגדרת במחלקה C, מתוך השיטה main? אם כן – כתוב את הזימון, אם לא – נמק מדוע.

22. נתונות שלוש המחלקות: NameA, NameB, NameC ומחלקה ראשית Test. עקוב בעזרת טבלת מעקב אחר ביצוע השיטה main במחלקה Test, ורשום את הפלט. על הטבלה לכלול את ערכי המשתנים, ובעבור כל עצם – את ערכי התכונות שלו.

```
public class NameA
{
    protected int x;
    public NameA(int x)
    { this.x = x; }
    public int getX()
    { return x; }
    public String toString()
    { return " NameA:x = "+ x ; }
    public String display()
    { return "display NameA"; }
}

public class NameB extends NameA
{
    protected int y;
    public NameB(int x , int y)
    {
        super(x);
        this.y = y;
    }
    public String toString()
    { return "NameB: x = "+ getX()+ "y = "+ this.y; }
}

public class NameC extends NameB
{
    protected int z;
    public NameC(int x , int y , int z)
    {
        super(x , y);
        this.z = z;
    }
    public String toString()
    { return " NameC: x = "+ x +" y = "+ y +" z = " + z; }
    public String display()
    { return "display NameC"; }
}
```



```

public class Test
{
    public static void main (String [] args)
    {
        NameC nc1 = new NameC(1 , 2 , 7);
        System.out.println (nc1.toString());
        NameA people [] = new NameA [4];
        people [0] = new NameA (6);
        people [1] = new NameB (8 , 9);
        people [2] = new NameC (4 , 0 , 11);
        people [3] = new NameC (3 , 2 , 4);
        people [3] = (NameA) people [2];
        int count = 0;
        for (int i=0; i < people.length; i++)
        {
            System.out.println (people [i]. display());
            if (people[i]. getX() > 5)
            {
                count ++;
            }
        }
        System.out.println(count);
        count = 0;
        for (int i=0; i < people.length; i++)
        {
            if (people[i] instanceof NameC)
            {
                count ++;
            }
        }
        System.out.println(count);
    }
}

```

23. חברת מוניות החליטה למחשב את ניהול המוניות שלה. לצורך כך הגדירו את המחלקה

מונית (Taxi), שלה התכונות האלה:

- מספר רישוי – `String taxiId`
- שם נהג – `String driverName`
- מספר מקומות לנוסעים (מלבד הנהג) – `int numPass`
- האם המונית פנויה – `boolean available`

התכונה `available` מאתחלת ל-`true`.

כמו כן הוגדרו פעולות האחזור:

`getId()`, `getName()`, `getNumPass()`, `isAvailable()`

**א.** ממש בג'אווה במחלקה `Taxi` את השיטות הבונות האלה:

i שיטה המקבלת כפרמטרים שם נהג ומספר רישוי,

מאתחלת בהתאמה את `driverName` ואת `taxiId`, ומאתחלת את `numPass`

להיות 4.

ii שיטה המקבלת כפרמטרים שם נהג, מספר רישוי, מספר מקומות לנוסעים,

ומאתחלת בהתאמה את `driverName`, `taxiId` ו-`numPass`.

**ב.** ממש בג'אווה במחלקה `Taxi` שיטה `taxiBusy` המעדכנת את המונית להיות

לא פנויה.

ג. בחברה רוצים להגדיר את המחלקה תחנת מוניות (TaxiStation), המשתמשת

במחלקה מונית (Taxi). לתחנה שייכות לכל היותר 80 מוניות.

התכונות של המחלקה TaxiStation הן:

- שם תחנה
- מספר מוניות השייכות לתחנה
- מערך של המוניות השייכות לתחנה

המוניות השייכות לתחנה מיוצגות במערך המוניות באופן רצוף מתחילת המערך.

ממש בג'אווה את המחלקה TaxiStation. מימוש המחלקה יכול רק את

הסעיפים i-iii שלפניך:

i הגדרת התכונות של המחלקה.

ii שיטה המוסיפה מונית לתחנה. השיטה מקבלת את שם הנהג, את מספר

הרישוי, ואת מספר המקומות לנוסעים.

הנח שמספר המוניות השייכות לתחנה קטן מ-80.

iii שיטה המקבלת מספר נוסעים, ומחזירה מספר רישוי של מונית פנויה כלשהי

המתאימה להסעת מספר נוסעים כמבוקש, ומעדכנת את המונית להיות לא

פנויה. אם אין מונית פנויה מתאימה, השיטה מחזירה -1.

24. המחלקה Rational מגדירה את טיפוס הנתונים **שבר** (מספר רציונלי). שבר מיוצג על ידי המונה שלו ועל ידי המכנה שלו.

לפניך חלק מהשיטות של המחלקה Rational:

Rational(int n1 , int n2)	שיטה בונה המחזירה שבר שהמונה שלו הוא n1 והמכנה שלו הוא n2
int getN1()	שיטה המאחזרת את המונה של שבר
int getN2()	שיטה המאחזרת את המכנה של שבר

המחלקה G מגדירה את טיפוס הנתונים **קבוצה**, שהוא אוסף עצמים מטיפוס Rational. מספר העצמים בקבוצה אינו קבוע. אין חשיבות לסדר הופעת העצמים בקבוצה.

לפניך חלק מהשיטות של המחלקה G:

G()	שיטה הבונה קבוצה ריקה
void gInsert(Rational num)	שיטה המכניסה לקבוצה את השבר num
Rational gRemove()	שיטה המחזירה שבר <u>כלשהו</u> הנמצא בקבוצה ומוחקת אותו מהקבוצה
boolean gEmpty()	שיטה המחזירה "אמת" אם הקבוצה ריקה, ו"שקר" אחרת

ממש בגיאוה, במחלקה G, את השיטות בסעיפים א-ב.

אפשר להשתמש בשיטות הנתונות של המחלקות Rational ו-G בלי לממש אותן.

G buildG(int n , int m)	שיטה המקבלת מספר שלם וחיובי n, ומספר שלם וחיובי m. השיטה מחזירה קבוצה המכילה את כל השברים שהמונה שלהם הוא מספר בין 1 ל-n (כולל), והמכנה שלהם הוא m.
-------------------------	---

א.

Rational sum(int m)	שיטה המקבלת מספר שלם m, ומחזירה שבר שהוא הסכום של השברים בקבוצה שהמכנה שלהם הוא m. הנח שיש בקבוצה לפחות שבר אחד שהמכנה שלו m. הערה: המונה של השבר המוחזר הוא סכום המונים של השברים האלה, והמכנה שלו הוא m.
---------------------	--

ב.

**שים לב**

השאלות בתכנית מונחה עצמים

לפותרים ב- C# החל מעמ' 30

שם לב: אם אתה כותב ב- C#, ענה על שתיים מהשאלות 25-28.

25. לפניך תכנית:

```
public class Stam
{
    public void B(int m , int n)
    { Console.WriteLine(m n); }
    public void f()
    {
        Console.WriteLine("inside Stam");
    }
}
public class B : Stam
    protected int y;
    public B(int x)
    {
        this.y = x;
    }
}
public class C : Stam
{
    public int x;
    public static void g()
    {
        Console.WriteLine ("inside C");
    }
}
public class Test
{
    public static void Main()
    {
        C temp = new C();
    }
}
```

ענה על הסעיפים א-ה, בהתייחס לתכנית הנתונה.

**א.** האם אפשר לשנות את הערך של התכונה  $x$  המוגדרת במחלקה  $C$ , מתוך שיטה במחלקה  $Stam$ ?

אם כן – כתוב כיצד, אם לא – נמק מדוע.

**ב.** i כתוב בשיטה  $Main$ , הוראה או הוראות לזימון השיטה  $B$  המוגדרת במחלקה  $Stam$ .

ii כתוב בשיטה  $Main$ , הוראה או הוראות לזימון השיטה  $B$  המוגדרת במחלקה  $B$ .

**ג.** לפניך שלוש הוראות, i-iii.

לכל אחת מהן, קבע אם אפשר להוסיף אותה בשיטה  $Main$ .

נמק את קביעותיך בעזרת מאפיינים של תכנות מונחה עצמים.

i  $Stam\ obj = new\ B(7);$

ii  $Stam\ obj = new\ B();$

iii  $B\ obj = new\ Stam();$

**ד.** האם אפשר לזמן את השיטה  $f()$  המוגדרת במחלקה  $Stam$ , מתוך שיטה במחלקה  $B$ ?

אם כן – כתוב את הזימון, אם לא – נמק מדוע.

**ה.** האם אפשר לזמן את השיטה  $g()$  המוגדרת במחלקה  $C$ , מתוך השיטה  $Main$ ?

אם כן – כתוב את הזימון, אם לא – נמק מדוע.

26. נתונות שלוש המחלקות: NameA, NameB, NameC ומחלקה ראשית Test. עקוב בעזרת טבלת מעקב אחר ביצוע השיטה Main במחלקה Test, ורשום את הפלט. על הטבלה לכלול את ערכי המשתנים, ובעבור כל עצם – את ערכי התכונות שלו.

```
public class NameA
{
    protected int x;
    public NameA(int x)
    { this.x = x; }
    public int X
    {
        get
        { return x; }
    }
    public override string ToString()
    { return "print NameA:x = " + x; }
    public virtual string Display()
    { return "display NameA"; }
}

public class NameB : NameA
{
    protected int y;
    public NameB(int x, int y):base (x)
    { this.y = y; }
    public override string ToString()
    { return "print NameB: x = " + x + "y = " + this.y; }
}

public class NameC : NameB
{
    protected int z;
    public NameC (int x , int y, int z):base (x , y)
    { this.z = z; }
    public override string ToString()
    { return "print NameC: x = " + x + " y = " + y + " z = " + z; }
    public override string Display()
    { return "display NameC"; }
}
```



```

public class Test
{
    public static void Main (string [] args)
    {
        NameC nc1 = new NameC(1 , 2 , 7);
        Console.WriteLine(nc1.ToString());
        NameA [] people = new NameA [4];
        people [0] = new NameA (6);
        people [1] = new NameB (8 , 9);
        people [2] = new NameC (4 , 0, 11);
        people [3] = new NameC (3 , 2 , 4);
        people [3] = (NameA) people [2];
        int count = 0;
        for (int i=0; i < people.Length; i++)
        {
            Console.WriteLine(people[i].Display());
            if (people[i].X > 5)
            {
                count++;
            }
        }
        Console.WriteLine(count);
        count = 0;
        for (int i=0; i < people.Length; i++)
        {
            if (people[i] is NameC)
            {
                count++;
            }
        }
        Console.WriteLine(count);
    }
}

```

27. חברת מוניות החליטה למחשב את ניהול המוניות שלה. לצורך כך הגדירו את המחלקה

מונית (Taxi), שלה התכונות האלה:

- מספר רישוי – `string taxiId`
- שם נהג – `string driverName`
- מספר מקומות לנוסעים (מלבד הנהג) – `int numPass`
- האם המונית פנויה – `bool available`

התכונה `available` מאתחלת ל-`true`.

כמו כן הוגדרו פעולות האחזור:

`GetId()` , `GetName()` , `GetNumPass()` , `IsAvailable()`

א. ממש ב-C# במחלקה `Taxi` את השיטות הבונות האלה:

i שיטה המקבלת כפרמטרים שם נהג ומספר רישוי,

מאתחלת בהתאמה את `driverName` ואת `taxiId`, ומאתחלת את `numPass`

להיות 4.

ii שיטה המקבלת כפרמטרים שם נהג, מספר רישוי, מספר מקומות לנוסעים,

ומאתחלת בהתאמה את `driverName`, `taxiId` ו-`numPass`.

ב. ממש ב-C# במחלקה `Taxi` שיטה `TaxiBusy` המעדכנת את המונית להיות

לא פנויה.

ג. בחברה רוצים להגדיר את המחלקה תחנת מוניות (TaxiStation), המשתמשת

במחלקה מונית (Taxi). לתחנה שייכות לכל היותר 80 מוניות.

התכונות של המחלקה TaxiStation הן:

- שם תחנה
- מספר מוניות השייכות לתחנה
- מערך של המוניות השייכות לתחנה

המוניות השייכות לתחנה מיוצגות במערך המוניות באופן רצוף מתחילת המערך.

ממש ב-C# את המחלקה TaxiStation. מימוש המחלקה יכלול רק את

הסעיפים i-iii שלפניך:

i הגדרת התכונות של המחלקה.

ii שיטה המוסיפה מונית לתחנה. השיטה מקבלת את שם הנהג, את מספר

הרישוי, ואת מספר המקומות לנוסעים.

הנח שמספר המוניות השייכות לתחנה קטן מ-80.

iii שיטה המקבלת מספר נוסעים, ומחזירה מספר רישוי של מונית פנויה כלשהי

המתאימה להסעת מספר נוסעים כמבוקש, ומעדכנת את המונית להיות לא

פנויה. אם אין מונית פנויה מתאימה, השיטה מחזירה -1.

28. המחלקה Rational מגדירה את טיפוס הנתונים **שבר** (מספר רציונלי). שבר מיוצג על ידי המונה שלו ועל ידי המכנה שלו.

לפניך חלק מהשיטות של המחלקה Rational:

Rational(int n1 , int n2)	שיטה בונה המחזירה שבר שהמונה שלו הוא n1 והמכנה שלו הוא n2
int GetN1()	שיטה המאחזרת את המונה של שבר
int GetN2()	שיטה המאחזרת את המכנה של שבר

המחלקה G מגדירה את טיפוס הנתונים **קבוצה**, שהוא אוסף עצמים מטיפוס Rational. מספר העצמים בקבוצה אינו קבוע. אין חשיבות לסדר הופעת העצמים בקבוצה.

לפניך חלק מהשיטות של המחלקה G:

G()	שיטה הבונה קבוצה ריקה
void GInsert(Rational num)	שיטה המכניסה לקבוצה את השבר num
Rational GRemove()	שיטה המחזירה שבר <u>כלשהו</u> הנמצא בקבוצה ומוחקת אותו מהקבוצה
bool GEmpty()	שיטה המחזירה "אמת" אם הקבוצה ריקה, ו"שקר" אחרת

ממש ב-C#, במחלקה G, את השיטות בסעיפים א-ב.

אפשר להשתמש בשיטות הנתונות של המחלקות Rational ו-G בלי לממש אותן.

G BuildG(int n , int m)	שיטה המקבלת מספר שלם וחיובי n, ומספר שלם וחיובי m. השיטה מחזירה קבוצה המכילה את כל השברים שהמונה שלהם הוא מספר בין 1 ל-n (כולל), והמכנה שלהם הוא m.
-------------------------	---

א.

Rational Sum(int m)	שיטה המקבלת מספר שלם m, ומחזירה שבר שהוא הסכום של השברים בקבוצה שהמכנה שלהם הוא m. הנח שיש בקבוצה לפחות שבר אחד שהמכנה שלו m. הערה: המונה של השבר המוחזר הוא סכום המונים של השברים האלה, והמכנה שלו הוא m.
---------------------	--

ב.

## ב ה צ ל ח ה !

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