

מדעי המחשב ה'**פתרון בחינת הגזרות****פרק א - עיצוב תכנה****שאלה 1:****פתרון בשפת Java:**

```

//--- פעולה המחזירה "אמת" אם הרשימה היא "רשימה משולשת" ו- "שקר" אחרת ---
//--- רשימה משולשת היא רשימה לא ריקה שמספר איבריה מתחלק ב- 3 ---
//--- והאיברים בכל שליש ברשימה זהים ומופיעים באותו סדר ---
public static boolean isTriple (List <Integer> list)
{
    int listSize = size(list);
    int n = listSize / 3;

    //--- אם הרשימה אינה עומדת בקריטריונים ---
    if (listSize == 0 || listSize % 3 != 0)
        return false;

    Node <Integer> pos1 = list.getFirst(); //פנייה לתחילת השליש הראשון
    Node <Integer> pos2 = place (list, n); //פנייה לתחילת השליש השני
    Node <Integer> pos3 = place (list, 2*n); //פנייה לתחילת השליש השלישי
    while (pos3 != null)
    {
        if(pos1.getInfo() != pos2.getInfo() || pos1.getInfo() != pos3.getInfo())
            return false;
        pos1 = pos1.getNext();
        pos2 = pos2.getNext();
        pos3 = pos3.getNext();
    }
    return true;
}

//--- פעולה המחזירה את מספר האיברים ברשימה ---
public static int size (List <Integer> list)
{
    int count = 0;
    Node <Integer> pos = list.getFirst();
    while (pos != null)
    {
        count ++;
        pos = pos.getNext();
    }
    return count;
}

//--- פעולה המקבלת רשימה ומספר n ---
//--- ומחזירה הפנייה לאיבר שנמצא במקום n ברשימה ---
//--- אם לא קיים איבר במקום זה יוחזר null ---
public static Node <Integer> place (List<Integer> list, int n)
{
    Node<Integer> pos = list.getFirst();
    while (pos != null && n > 0)
    {
        n -- ;
        pos = pos.getNext();
    }
    return pos;
}

```

פתרון בשפת C# - נכתב ע"י ראמי גבאלי:

```
//פעולה מקבלת רשימה ומחזירה את מספר האיברים ברשימה
public static int Count(List<int> list)
{
    Node<int> p = list.GetFirst();
    int c = 0;
    while (p != null)
    {
        c++;
        p = p.GetNext();
    }
    return c;
}

//פעולה מקבלת רשימה ומחזירה אמת אם הרשימה היא משולשת
public static bool IsThree(List<int> list)
{
    int n=Count(list);
    if (!list.IsEmpty() && n % 3 == 0)
    {
        Node<int> p = list.GetFirst();
        Node<int> q = p;
        for (int i = 1; i <= n / 3; i++)
            q = q.GetNext();
        while (q != null)
        {
            if (p.GetInfo() != q.GetInfo())
                return false;
            p = p.GetNext();
            q = q.GetNext();
        }
        return true;
    }
    else return false;
}
```

פתרון בשפת C# - נכתב ע"י דיתה אוהב ציון:

```
// טענת קלט : פעולה המקבלת רשימה
// טענת פלט : מחזירה את אורך הרשימה
public static int GetLength(List<int> l)
{
    Node<int> pos = l.GetFirst();
    int counter = 0;
    while (pos != null)
    {
        counter++;
        pos = pos.GetNext();
    }
    return counter;
}

// ענת קלט : פעולה המקבלת רשימה ומספר המיצג מיקום ברשימה
// טענת פלט : הפניה לחוליה שהמספר הוא מיקומה הסידרתי ברשימה
public static Node<int> GetPos(List<int> l, int place)
{
    Node<int> pos = l.GetFirst();
    for (int i = 0; i < place; i++)
        pos = pos.GetNext();
    return pos;
}
```

הפעולה הנדרשת:

```
public static bool ThreeList(List<int> l)
{
    int listSize = GetLength(l);
    if (listSize == 0 || listSize % 3 != 0)
        return false;
    int part = listSize / 3; // גודל כל חלק

    bool status = true;
    Node<int> pos1 = l.GetFirst(); // הפניה לשליש הראשון
    Node<int> pos2 = GetPos(l, part); // הפניה לשליש השני
    Node<int> pos3 = GetPos(l, part * 2); // הפניה לשליש השלישי
    while (status && pos3 != null) // הסריקה
    {
        if(pos1.GetInfo() != pos2.GetInfo() || pos2.GetInfo() != pos3.GetInfo())
            status = false;
        pos1 = pos1.GetNext();
        pos2 = pos2.GetNext();
        pos3 = pos3.GetNext();
    }
    return status;
}
```

שאלה 2:

א. עץ-ימין-שמאל (t)

אם $t == \text{null}$ או עלה? (t)

החזר אמת

אם ל-t יש בן ימני ואין לו בן שמאלי

החזק שקר

החזר עץ-ימין-שמאל (בן שמאלי של t) וגם עץ-ימין-שמאל (בן שמאלי של t)

פתרון בשפת Java:

--- פעולה המחזירה "אמת" אם העץ הוא עץ ימין-שמאל ו-"שקר" אחרת
 --- עץ ימין-שמאלי הוא עץ בינארי שבו לכל צומת, אם קיים בן ימני אז קיים גם בן שמאלי ---

```
public static boolean leftRightTree (BinTreeNode <Integer> t)
{
    if (t == null || isLeaf(t))
        return true;

    if (t.getRight() != null && t.getLeft() == null)
        return false;

    return leftRightTree (t.getLeft()) && leftRightTree (t.getRight());
}

--- פעולה המחזירה "אמת" אם העץ הוא עלה, ו-"שקר" אחרת ---
public static boolean isLeaf (BinTreeNode <Integer> t)
{
    return t.getLeft() == null && t.getRight() == null;
}
```

פתרון בשפת C# - נכתב ע"י ראמי גבאלי:

```
// פעולה מחזירה אמת אם הצומת היא עלה ושקר אחרת
public static bool IsLeaf(BinTreeNode<int> bt)
{
    return (bt.GetLeft() == null) && (bt.GetRight() == null);
}

// פעולה מחזירה אמת אם העץ הוא ימין-שמאל
public static bool IsRightLeft(BinTreeNode<int> t)
{
    if (t == null || IsLeaf(t))
        return true;
    if (t.GetRight() != null)
        if (t.GetLeft() == null)
            return false;
    return IsRightLeft(t.GetLeft()) && sRightLeft(t.GetRight());
}
```

פתרון בשפת C# - נכתב ע"י דיתה אוהב ציון:

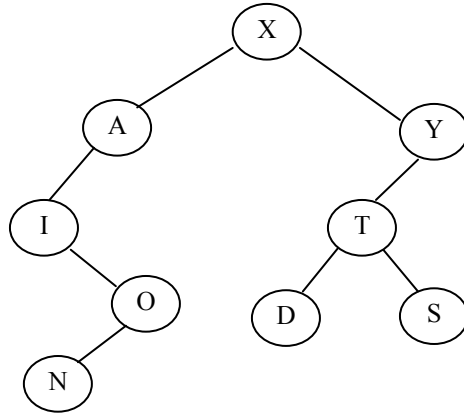
```
פעולת עזר : האם עלה ? מחזירה אמת אם ההפניה היא עלה.
public static bool isLeaf(BinTreeNode<int> t)
{
    if (t.GetRight() == null && t.GetLeft() == null)
        return true;
    else
        return false;
}

public static bool LeftRightTree(BinTreeNode<int> t)
{
    if (isLeaf(t))
        return true;
    else
        if (t.GetRight() != null && t.GetLeft() == null) // אם לבן ימני אין אחר
            return false;
        else
            if (t.GetRight() != null && t.GetLeft() != null) // שני בנים
                return LeftRightTree(t.GetRight()) &&
                    LeftRightTree(t.GetLeft());
            else
                if (t.GetRight() != null)
                    return LeftRightTree(t.GetRight());
                else
                    return LeftRightTree(t.GetLeft());
}
```

ב.

PreOrder: X A I O N Y T D S

InOrder: I N O A X D T S Y

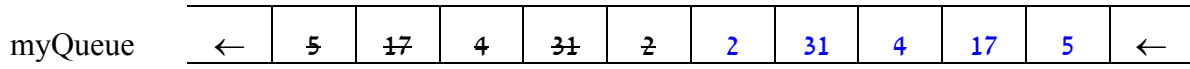


PostOrder: N O I A D S T Y X

שאלה 3:

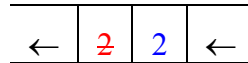
א. `sod1(myQueue)`

א. מעקב Sod1

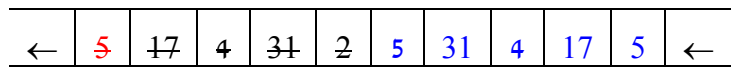
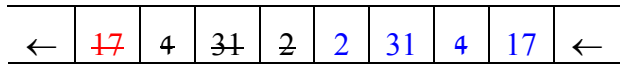
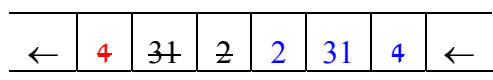
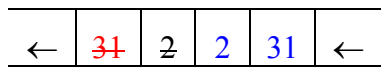


טבלת מעקב, לפי מודל "האנשים הקטנים" נוצרת מחסנית של זימונים רקורסיביים. כל "איש קטן" שסיים עבודתו, מחזיר ערך לאיש שמתחתיו ויוצא מהמחסנית.

E	$i = 2$ $result = 2$ $!q.isEmpty \rightarrow F$ ערך מוחזר 2
D	$i = 31$ $result = 31$ $j = ? = 2$ $result > j \rightarrow T$ ערך מוחזר 2
C	$i = 4$ $result = 4$ $j = ? = 2$ $result > j \rightarrow T$ ערך מוחזר 2
B	$i = 17$ $result = 17$ $j = ? = 2$ $result > j \rightarrow T$ ערך מוחזר 2
A	$i = 5$ $result = 5$ $j = ? = 2$ $result > j \rightarrow T$ ערך מוחזר 2



זימון אחרון



זימון ראשון

הפעולה `sod1` מחזירה את המספר 2 (תוך כדי כך היא הופכת את סדר האיברים בתור).

ב. הפעולה מחזירה את המספר (האיבר) הקטן ביותר בתור.

ג. sod2 (17852)

	i	i == 0	a	ערך מוחזר ← b	a > b	משפט זימון
↓	17852	F	2	8	F	sod2(1785)
	1785	F	5	8	F	sod2(178)
	178	F	8	7	T	sod2(17)
	17	F	7	1	T	sod2(1)
	1	F	1	0	T	sod2(0)
	0	T				

הפעולה sod1 מחזירה את המספר 8.

ד. הפעולה מחזירה את הספרה הגדולה ביותר במספר.

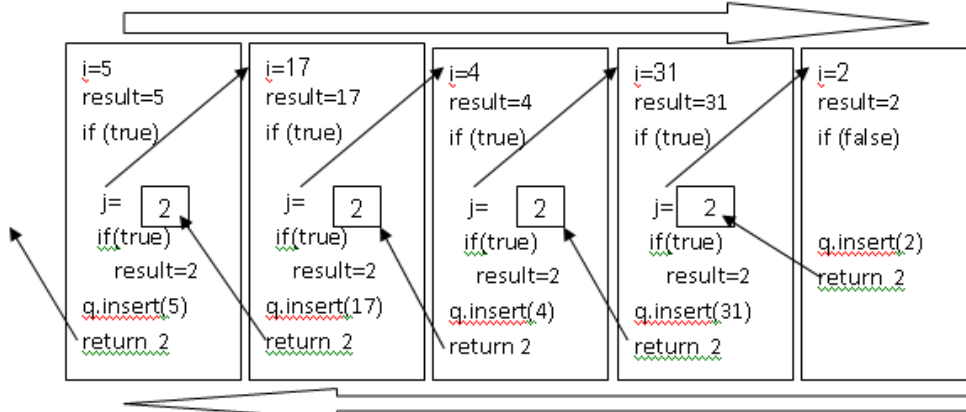
ה. הפעולה מחזירה את הספרה הגדולה ביותר במספר הקטן ביותר בתור.

פתרון נוסף - נכתב ע"י דיתה אוהב ציון:

א. מעקב Sod1

ראש התור	5	17	4	31	2	סוף התור
----------	---	----	---	----	---	----------

בקריאה הרקורסיבית התור מתחלקן



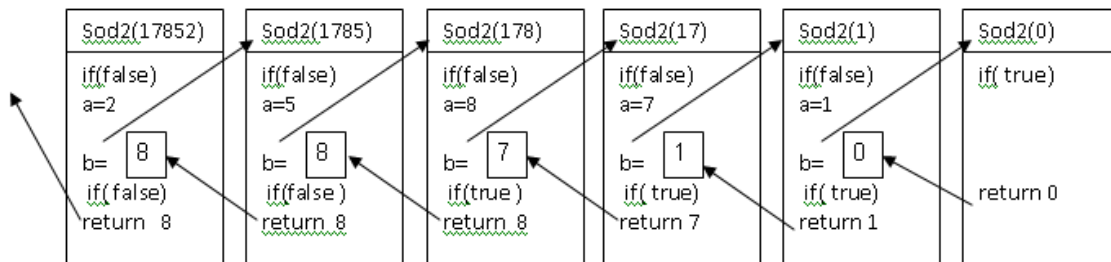
בחזרה התור מתמלא מחדש בסדר הפוך

ראש התור	2	31	4	17	5	סוף התור
----------	---	----	---	----	---	----------

הפעולה תחזיר את המספר 2.

ב. הפעולה מחזירה את המספר הקטן ביותר נמצא בתור, והופכת את סדר התור.

ג. מעקב Sod2



הפעולה תחזיר את המספר 8.

ד. הפעולה תחזיר את הספרה הגדולה ביותר במספר k.

ה. הפעולה תחזיר את הספרה הגדולה ביותר במספר הקטן ביותר בתור.

שאלה 4:

פתרון בשפת Java:

א. כותרת המחלקה DayInSchedule :

```
//--- DayInSchedule המחלקה
public class DayInSchedule
{
    private int day;
    private int month;
    private List <Meeting> mList;
```

ב. מימוש הפעולה canStart :

```
//--- פעולה המחזירה "אמת" אם ניתן לקבוע פגישה המתחילה בשעה startHour ---
//--- ומסתיימת אחרי minutes דקות ---
public boolean canStart (int startHour, int minutes)
{
    List <Integer> freeHour = getFreeHours();
    Node<Integer> pos = freeHour.getFirst();

    //--- בדיקה - האם פנוי בשעת תחילת הפגישה ---
    while (pos != null && pos.getInfo() < startHour)
        pos = pos.getNext();
    if (pos == null || pos != null && pos.getInfo() > startHour)
        return false;

    //--- קביעת שעת סיום הפגישה ---
    int finishHour = startHour + minutes/60;
    if (minutes % 60 > 0)
        finishHour ++;

    //--- בדיקה - האם פנוי למשך כל זמן הפגישה ---
    int nextHour = startHour + 1;
    pos = pos.getNext();
    while (pos != null && nextHour < finishHour)
    {
        if (pos.getInfo() != nextHour)
            return false;
        nextHour ++;
        pos = pos.getNext();
    }
    return true;
}
} //--- end of class DayInSchedule
```

ג. הפעולה printAvailable :

```
//--- פעולה המקבלת את יומן הפגישות ופגישה ---
//--- ומציגה כפלט את כל הימים בהם ניתן לשלב הפגישה ---
public static void printAvailableDay (List <DayInSchedule> lst, Meeting m)
{
    Node<DayInSchedule> pos = lst.getFirst();
    while (pos != null)
    {
        if (pos.getInfo().canStart(m.getStartHour(), m.getMinutes()))
            System.out.println(pos.getInfo().getDay() + "/" +
                               pos.getInfo().getMonth());
        pos = pos.getNext();
    }
}
```

פתרון בשפת C# - נכתב ע"י ראמי גבאלי:

מחלקה DayInSchedule

```
public class DayInSchedule
{
    private int day;
    private int month;
    private List<Meeting> listMetting;

    //פעולה מחזרה אמת אפשר להתחיל פגישה בשעה המבוקשת/
    public bool CanStart(int startHour, int minutes)
    {
        List<int> freeHours = GetFreeHours();
        Node<int> pos = freeHours.GetFirst();
        int time = minutes;
        if (time % 60 != 0)
            time = (time / 60) + 1;
        else time = time / 60;
        time = time + startHour;
        while (pos != null && pos.GetInfo() < startHour)
            pos = pos.GetNext();
        if (pos == null || pos.GetInfo() > startHour)
            return false;
        int nexthour = startHour + 1;
        pos = pos.GetNext();
        while (pos != null && nexthour < time)
        {
            if (pos.GetInfo() != nexthour)
                return false;
            nexthour++;
            pos = pos.GetNext();
        }
        return true;
    }
}

//פעולה מדפיסה את החודש והיום בחודש שבהם אפשר לשבץ פגישות/
public static void PrintAvailbleDay
    (List<DayInSchedule> lst, Meeting m)
{
    Node<DayInSchedule> p = lst.GetFirst();
    while (p != null)
    {
        if(p.GetInfo().CanStart(m.GetStartHour(),m.GetMinutes()))
        {
            Console.WriteLine(p.GetInfo().GetDay());
            Console.WriteLine(p.GetInfo().GetMoth());
        }
        p = p.GetNext();
    }
}
```

פתרון בשפת C# - נכתב ע"י דיתה אוהב ציון:

א.

Class DayInSchedule
<pre>int day; int month; List<Meeting> meet; //רשימת פגישות מתוכננות</pre>

ב. הפעולה CanStart

```
public bool CanStart(int startHour, int minutes)
{
    List<int> freeH = GetFreeHours(); // רשימת שעות פנויות
    bool found = false;
    Node<int> pos = freeH.GetFirst();
    int hourLength;
    while (pos != null && !found) // טריקת הרשימה
    {
        if (pos.GetInfo() == startHour)
        {
            if (minutes < 60)
                found = true;
            else
            {
                if (minutes % 60 == 0) // חישוב משך הפגישה בשעות
                    hourLength = minutes / 60;
                else
                    hourLength = minutes / 60 + 1;

                pos = pos.GetNext();
                for (int i = 0; i < hourLength && pos != null; i++)
                {
                    startHour++;
                    if (pos.GetInfo() == startHour)
                        found = true;
                    else
                        found = false;
                    pos = pos.GetNext();
                }
            }
        }
    }
    return found;
}
```

ג. פעולה חיזונית

```
public static void PrintAvailableDay(List<DayInSchedule> lst, Meeting m)
{
    Node<DayInSchedule> pos = lst.GetFirst();
    while (pos != null)
    {
        if (pos.GetInfo().CanStart(m.GetStartHour(), m.GetMinuites()))
        {
            Console.WriteLine("In month {0} day {1} ",
                pos.GetInfo().GetMonth(), pos.GetInfo().GetDay());
        }
        pos=pos.GetNext();
    }
}
```

פרק ב'

מערכות מחשב ואסמבלר

הפתרון לפרק זה נכתב ע"י מיקדאד עלי-סאלח

תרגיל 5:

א. i מצב האוגרים

SI	DL	AX		CX	DX	BX
		AH	AL			
0000	02	00	05	0003	0002	0002
0001		00	03	0002	0001	0004
0003		00	02	0001	0000	0003
0004		00	03	0000	0003	0006
		00	04		0002	0009
		00	05		0001	0004
					0000	0008
					0004	0012
					0003	0016
					0002	
					0001	
					0000	

מצב מקטע הזיכרון:

שם משתנה	NUM1	NUM2	ARR					
כתובת	0000	0001	0002	0003	0004	0005	0006
תוכן	02	05	04	09	16	00	?	?

ii. התוכנית מציבה במערך ARR (משמאל לימין):

$(NUM1)^2$	$(NUM1+1)^2$	$(NUM1+2)^2$	$(NUM1+3)^2$
------------	--------------	--------------	--------------	-------

(NUM2 - NUM1) איברים

- ב. i. אינו מבצע את הנדרש
- ii. אינו מבצע את הנדרש
- iii. מבצע את הנדרש
- iv. אינו מבצע את הנדרש

תרגיל 6:

MOV AH , 0	.i	א.
JZ A2	.ii	
ADD DX , 2	.iii	
MOV AX , BX	.iv	
מבצע את הנדרש	.i	ב.
מבצע את הנדרש	.ii	
אינו מבצע את הנדרש	.iii	
אינו מבצע את הנדרש	.iv	

תרגיל 7:

```

LEA SI , ARR1
LEA DI , ARR2
MOV CX , 36
AGAIN1:  PUSH CX
         MOV BX , SI
         MOV AH , [BX]
AGAIN2:  CMP BYTE PTR [BX] , 0
         JE  NEXT1
         CMP [BX] , AH
         JNE NEXT1
         MOV BYTE PTR [BX] , 0
NEXT1:   INC  BX
         LOOP AGAIN2
         CMP AH , 0
         JE  NEXT2
         MOV [DI] , AH
         INC DI
NEXT2:   POP  CX
         INC SI
         LOOP AGAIN1
    
```

שאלה 8:

```
MOV ANS , 1
MOV AH , 1
AGAIN1: MOV AL , 0
        LEA SI , ARR
        MOV CX , 58
AGAIN2: CMP [SI] , AH
        JNE NEXT
        INC AL
NEXT:   INC SI
        LOOP AGAIN2
        CMP AH , AL
        JNE SOF
        INC AH
        CMP AH , K
        JBE AGAIN1
        JMP FINISH
SOF:   MOV ANS , 0
FINISH: NOP
```

פרק ב'
מבוא לחקר ביצועים

שאלה 9:

שאלה 10:

שאלה 11:

שאלה 12:

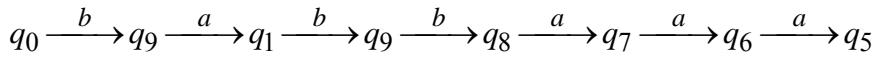
פרק ב'

מודלים חישוביים

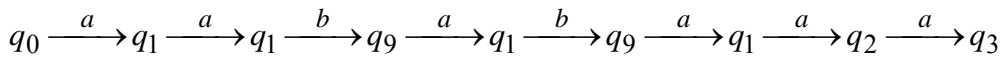
הפתרון לפרק זה נכתב ע"י רחל לודמר.

תרגיל 13:

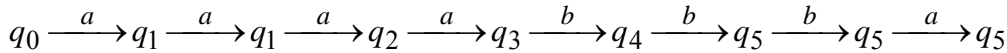
א. (i) (1) המילה babbaaa מתקבלת.



(2) המילה aababaaa לא מתקבלת.



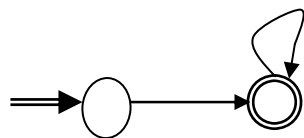
(3) המילה aaaabbbba מתקבלת.



(ii) המילים הקצרות ביותר המתקבלות ע"י האוטומט: bbaaa, aaabb.

ב. בשפה L_1 ה- b בא אחרי ה- a, ובשפה L_2 ה- b בא לפני ה- a. לכן בחיתוכם מספר ה- b חייב להיות 0.

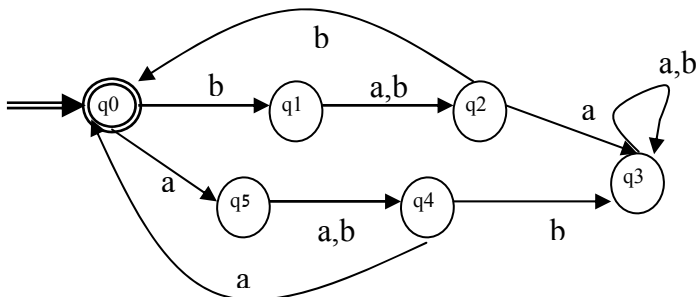
$$\begin{aligned} L_1 \cap L_2 &= \{a^n b^k \mid n \neq k, n \geq 0, k \geq 0\} \cap \{b^i a^j \mid i \geq 0, j \geq 1\} \\ &= \{a^n b^k \mid n > 0, k = 0\} \cap \{b^i a^j \mid i = 0, j \geq 1\} \\ &= \{a^n \mid n > 0\} \cap \{a^j \mid j \geq 1\} = \{a^n \mid n \geq 1\} \end{aligned}$$



החיתוך הוא שפה רגולרית. נבנה אוטומט מתאים.

שאלה 14:

א.



ב. $L \cap \{(aab)^n \mid n \geq 0\} = \{w^j \mid j \geq 0, w = \{\epsilon, aba, bab, bbb, aaa\} \cap \{(aab)^n \mid n \geq 0\} = \{\epsilon\}$

החיתוך הוא השפה המכילה את המילה הריקה.

שאלה 15:

א.
$$\begin{aligned} L_1 \cap L_2 &= \{0^n 1^n 0^k 1^k \mid n \geq 1, k \geq 0\} \cap \{0^n 1^k \mid n \geq 0, k \geq 0\} \\ &= \{0^n 1^n \mid n \geq 1\} \cap \{0^n 1^k \mid n \geq 0, k \geq 0\} \\ &= \{0^n 1^n \mid n \geq 1\} \end{aligned}$$

ב. $Init(L)$ היא קבוצת כל ההתחלות של המילים בשפה L .
 כל מילה ב- L ניתן לפרק אותה לשרשור של 2 מילים. הרישא של כל מילה תהיה ב- $Init(L)$.

$$\begin{aligned} Init(L_3) &= \{u \mid uv \in L_3, u, v \in \Sigma^*\} \\ &= \{0^i 1^{i-k} \mid 0^i 1^{i-k} \cdot 1^k \in L_3, i \geq 0, 0 \leq k \leq i\} \quad u = 0^i 1^{i-k} \quad 0 \leq k \leq i, v = 1^k \end{aligned}$$

ג. $Fin(L)$ היא קבוצת כל הסייפות של המילים בשפה L .
 באופן דומה לסעיף הקודם, כל מילה ב- L ניתן לפרק אותה לשרשור של 2 מילים. הסייפא של כל מילה תהיה ב- $Fin(L)$.

$$\begin{aligned} Fin(L_3) &= \{v \mid uv \in L_3, u, v \in \Sigma^*\} \\ &= \{1^k \mid 0^i 1^{i-k} \cdot 1^k \in L_3, i \geq 0, 0 \leq k \leq i\} \quad u = 0^i 1^{i-k} \quad 0 \leq k \leq i, v = 1^k \end{aligned}$$

פתרון נוסף לסעיף זה: נכתב ע"י יבגני קנל

$$\begin{aligned} Fin(L_3) &= \{v \mid uv \in L_3, u, v \in \Sigma^*\} \\ &= \{0^{i-k} 1^i \mid 0^k \cdot 0^{i-k} 1^i \in L_3, i \geq 0, 0 \leq k \leq i\} \quad v = 0^{i-k} 1^i \quad 0 \leq k \leq i, u = 0^k \end{aligned}$$

ד. האם $0011 \in Min(L_4)$?

$Min(L_4)$ מכילה את אוסף המילים ב- L_4 המקיימות את התנאי הבא:
 עבור כל פירוק של מילה ב- L_4 , $w = w_1 \cdot w_2$, שהסייפא שלה (w_2) לא ריקה,
 הרישא שלה (w_1) לא שייכת לשפה L_4 .

נסתכל על השרשורים שיוצרים את 0011, ללא סייפא ריקה.

$$0011 = \{\varepsilon \cdot 0011, 0 \cdot 011, 00 \cdot 11, 001 \cdot 1\}$$

השרשור $\varepsilon \cdot 0011$ לא מקיים את התנאי שהרישא שלו, לא שייכת ל- L_4 . ($\varepsilon \in L_4$).
 לכן $0011 \notin Min(L_4)$.

(שאר השרשורים כן מקיימים: $00 \notin L_4 \quad i=2, k=0, i > k$, $0 \notin L_4 \quad i=1, k=0, i > k$, $001 \notin L_4 \quad i=2, k=1, i > k$)

ה. האם $0011 \in \text{Min}(L_5)$? ההסבר זהה לסעיף קודם.

$\text{Min}(L_5)$ מכילה את אוסף המילים ב- L_5 המקיימות את התנאי הבא:
 עבור כל פירוק של מילה ב- L_5 , $w = w_1 \cdot w_2$, שהסייפא שלה (w_2) לא ריקה,
 הרישא שלה (w_1) לא שייכת לשפה L_5 .

נסתכל על השרשורים שיוצרים את 0011, ללא סייפא ריקה.

$$0011 = \{\varepsilon \cdot 0011, 0 \cdot 011, 00 \cdot 11, 001 \cdot 1\}$$

אף לא אחד מהפירוקים של 0011, לא מקיים את התנאי שהרישא שלו, לא שייכת ל- L_5 :

$$\varepsilon \in L_5$$

$$0 \in L_5 \quad i = 1, k = 0, k < i$$

$$00 \in L_5 \quad i = 2, k = 0, k < i$$

$$001 \in L_5 \quad i = 2, k = 1, k < i$$

לכן $0011 \notin \text{Min}(L_5)$. (מספיק להראות שהרישא של אחד מהשרשורים שייכת ל- L_5)

ו. האם השפה $L_4 \cap L_5$ היא רגולרית?

$$\begin{aligned} L_4 \cap L_5 &= \{0^i 1^k \mid 0 \leq i \leq k\} \cap \{0^i 1^k \mid 0 \leq k \leq i\} \\ &= \{0^i 1^i \mid i \geq 0\} \end{aligned}$$

שפת החיתוך אינה רגולרית. בשפה זו יש תלות של מניה בין רצף ה-0 לבין רצף ה-1.

שאלה 16:

⊥	1	1	1	1	1	1	1	\$	Δ
---	---	---	---	---	---	---	---	----	---

א. לאחר חישוב $f(5)$ הסרט יכיל:

⊥	1	1	1	1	1	1	1	\$	Δ
---	---	---	---	---	---	---	---	----	---

ב. לאחר חישוב $f(6)$ הסרט יכיל:

ג. מטרת הפונקציה: $f(x) = x + x \% 2$.

פרק ב'

תכנות מונחה עצמים Java

תרגיל 17:

- א. static - משתנה של המחלקה. קיים פעם אחת והוא משותף לכל האובייקטים של המחלקה.
 אם הוא מוגדר כ- public ניתן לשנותו מכל מקום: ; Polygon.maxSides = חדש
 ב. maxSides הינו קבוע של המחלקה. לא ניתן לשנותו לאחר האתחול.
 ג. (i) פעולה פנימית במחלקה Polygon:

```
public void expand (Polygon poly)
{
    (*) if (this.numSides + poly.numSides < maxSides)
        {
            for (int i = 0 ; i < poly.numSides ; i++)
                addSide (poly.values[i]);
        }
}
```

(*) ללא ההנחה שיש מספיק מקום, יש להוסיף בדיקה שיש מספיק מקום.

(ii) פעולה חיצונית:

```
public static void expand (Polygon poly1, Polygon poly2)
{
    int num = poly2.getNumSides();
    if (poly1.getNumSides() + num < Polygon.maxSides)
    {
        for (int i = 0 ; i < num ; i++)
            poly1.addSide (poly2.getValues()[i]);
    }
}
```

ג.

- (i) p1.expand (p2);
- (ii) expand (p1, p2);
- (iii) PolygonOperations.expand (p1, p2);

תרגיל 18:

```
class Buy
{
    private int price;           // מחיר הקניה
    private Date date;          // תאריך הקניה
    private Payment [ ] payments; // מערך התשלומים
    private int numOfPayments;  // מס' תשלומים

    // פעולה המחזירה "אמת" אם סה"כ התשלומים שווה לסכום הקניה, ו- "שקר" אחרת
    public boolean checkPayment ( )

    public void printReceipt ( ) // הדפסת קבלה בעבור הקניה
}

public class Store
{
    private Buy [ ] purchase;    // כלל הרכישות בחנות
    private int count;          // מספר הרכישות שהתבצעו

    // פעולה המקבלת תאריך, סכום רכישה ואמצעי התשלום ומוסיפה רכישה למערך הרכישות
    public void addPurchase (Date date, int price, Payment [ ] payments);

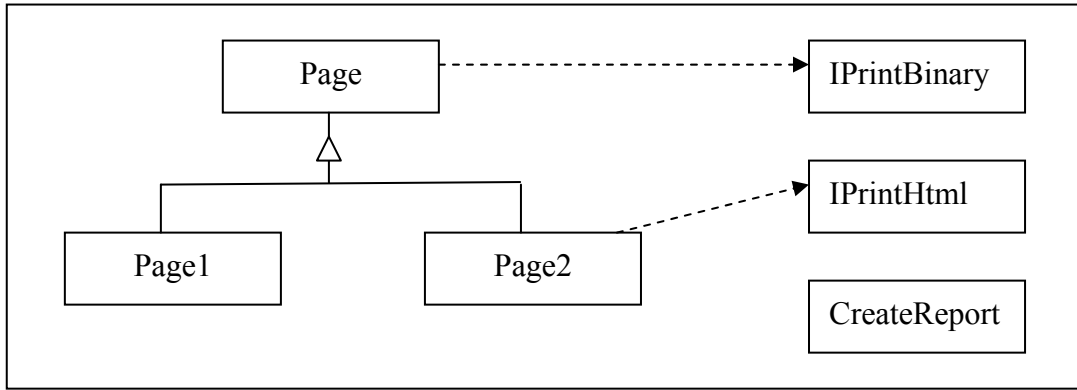
    public int dayTotalSum (Date date) // חישוב סכום הקניות של יום מסוים
}

public class Payment
{
    protected int sumToPay;     // הסכום לתשלום
}

public class Cash extends Payment { ... }

class Cheque extends Payment
{
    private int chequeNum;      // מס' ההמחאה
    private String bankName;    // שם הבנק
    private Date chequeDate;    // תאריך ההמחאה
}

class Credit extends Payment
{
    private int creditCardNum;  // מס' כרטיס האשראי
    private Date expiryDate;    // תאריך התפוגה של כרטיס האשראי
    private Date debitDate;     // תאריך החיוב
}
```



תרגיל 19:

- א. (i) פעולה לא חוקית.
 doc1 הוא אובייקט מסוג Page1 שיוורש מ- Page שמממש את IPrintBinary.
 אין קשר בין page1 ו- IPrintHtml ולכן לא ניתן להעביר את doc1 לפעולה CreateHtmlDoc שבמחלקה CreateReport.
- (ii) פעולה לא חוקית.
 ההפניה היא מסוג המחלקה Page שאינה מממשת את IPrintHtml ולכן תהיה שגיאת קומפילציה.
- (iii) פעולה חוקית.
 ההפניה היא מסוג IPrintBinary והיא מפנה לאובייקט מסוג Page1 היוורש מ- Page המממש את IPrintBinary. פעולה מצפה לקבל אובייקט מסוג IPrintBinary.
- ב. גם אם doc2 הוא מופע של Page2, ההפניה אליו היא מסוג Page שאינה מממשת את הפעולה createHtml.
 כדי שהפעולה תהיה חוקית, יש לבצע המרה כלפי מטה: ((Page2)doc).createHtml()

ג.

```

*** Html Doc ***
Html Data : num = 11 , num1 = 22
11
22
*** Binary Doc ***
Binary Data : num = 33 , num1 = 44
33
44
*** Binary Doc ***
55
66
    
```

שאלה 20:

```
public class Park
{
    private String parkName;           // שם גן השעשועים
    private Attraction [] attractions;  // מערך האטרקציות
    private int numOfAttractions;      // מס' האטרקציות ברגע הנוכחי

    public Park (String name, int maxAttractions)
    {
        this.parkName = name;
        this.attractions = new Attraction [maxAttractions];
        this.numOfAttractions = 0;
    }

    public void printShowsWithTranslation( )
    {
        for (int i = 0 ; i < numOfAttractions ; i ++ )
            if (attractions[i] instanceof Show)
                if ((Show)attractions[i].getTranslation())
                    System.out.println (attractions[i].getName());
    }

    public static void main (String [] args)
    {
        Park lunafun = new Park ("lunafun", 30);
        Attraction a = new Ride ("MountainTrain", 25,12, 3, 1);
        lunafun. addAttraction (a);
        a = new Show ("Bambi", 150, 3, "Children", true);
        lunafun. addAttraction (a);
    }
}
```

ג.

פרק ב'תכנות מונחה עצמים C#

הפתרון לפרק זה נכתב ע"י טובי סטפ

תרגיל 21:

- א. המשמעות הוא שהנתון maxSides מוגדר ברמת המחלקה והיא משותפת לכל ה-Polygons (כל Polygon יכול להיות לו עד 100 צלעות ואינו תלוי באובייקט) זימון ניתן לעשות ברמת המחלקה.
- ב. המשמעות שהוא מוגדר כ-const משמעותו שלא ניתן לשנות את הערך לאחר האתחול.
- ג. (i) פעולה פנימית במחלקה Polygon

```
public void Expand (Polygon p)
{
    if (p.numSides + this.numSides <= Polygon.maxSides)
    {
        for (int i= 0; i < p.numSides; i++)
            addSides( p.values[i]);
    }
}
```

(ii) פעולה סטטית חיצונית:

```
public static void Expand (Polygon p1, Polygon p2)
{
    if (p1.getNumSides() + p2.getNumSides() <= Polygon.maxSides)
    {
        for (int i = 0; i < p2.getNumSides(); i++)
            p1.addSide (p2.getValues( ) [i]);
    }
}
```

.ד

- i p1.Expand (p2);
- ii Expand (p1,p2);
- iii PolygonOperations.Expand (p1, p2);

תרגיל 22:

```
class Cash : Payment {
}

class Credit:Payment { // מחלקת כרטיס אשראי
    private int creditCardNumber; // מס' כרטיס אשראי
    private Date expiryDate; // תאריך תפוגה של כרטיס אשראי
    private Date debitDate; // תאריך חיוב
}

class Cheque:Payment {
    private int chequeNumber; // מס' הצ'ק
    private string bankName; // שם הבנק
    private Date chequeDate; // תאריך הצ'ק
}

class Payment {
    private int sum; // סכום לתשלום
}

class Buy {
    private int total; // סה"כ רכישה
    private Date purchaseDate; // תאריך רכישה
    private Payment[] payments; // תשלומים שונים
    private int count; // מס' תשלומים

    public bool Check () // פעולה הבודקת האם סה"כ התשלומים שווה לסה"כ
    public void PrintReceipt ()
}

class Store {
    private Buy[] purchases; // אוסף של קניות החנות
    private int count; // מספר הקניות

    // שיטה המקבלת תאריך הרכישה, סכום לתשלום ואמצעי התשלום
    public void AddPurchase (Date day, int total, Payment[] payments);
    public int DaySum (Date date) // חישוב סכום הקניות של יום מסוים
}
}
```

תרגיל 23:

א.

i. לא חוקי. doc1 הוא אובייקט מסוג Page1. Page1 יורש מ- Page שירש מ- IPrintBinary. אין הורשה מ- IPrintHtml ולכן לא ניתן להעביר אותו כפרמטר לפעולה CreateReport.createHtmlDoc

ii.

לא חוקי. מחלקת Page אינו יורש מ- IPrintHtml ולכן זה לא יעבור קומפילציה.

iii.

נכון. הוא שומר אובייקט מסוג Page1 שירש מ- Page המממש IPrintBinary. זימון הפעולה IPrintBinary מקבל פרמטר מסוג CreateReport.createBinaryDoc

ב.

השגיאה תהיה ב- doc.createHtml() כי לאובייקט Page אין פעולה createHtml ולכן יש לבצע המרה כלפי מטה ((Page2)doc).createHtml()

ג.

*** Html Doc ***

Html Data : num = 11 , num1 = 22

11

22

*** Binary Doc ***

Binary Data : num = 33 , num1 = 44

33

44

*** Binary Doc ***

55

66

שאלה 24:

```
class Park{
    private string parkName;           // שם גן השעשועים
    private int currentNumberOfAttractions; // מס' האטרקציות כרגע
    private Attraction [] attractions ; // האטרקציות הקיימות

    public Park (string name, int maxAttractions)
    {
        this.parkName = name;
        attractions = new Attraction [maxAttractions];
        currentNumberOfAttractions = 0;
    }

    public void PrintShowsWithTranslation()
    {
        for (int i=0; i < currentNumberOfAttractions; i++)
            if (attractions[i] is Show)
                if ((Show)attractions[i].GetTranslation())
                    Console.WriteLine (attractions[i].GetName());
    }

    public static void Main (string[] args)
    {
        Park lunafun = new Park ("lunafun", 30);
        Attraction a = new Ride ("DareDevil", 10,14,5, 1);
        lunafun.AddAttraction (a);
        a = new Show ("Snow White", 100, 0, "Childen", true);
        lunafun. AddAttraction (a);
    }
}
```

.ג