

מדעי המחשב

פרק ראשון

שאלה 1

Java

```
//--- פעולה המחזירה מערך חדש שבו כל האיברים ---
//--- שלא נמצאים במערך הנתון ---
public static int[] Filter (int[]arr, int num)
{
    int n = countNums(arr, num);
    int[] a = new int[n];
    int j = 0;
    for(int i = 0; i < arr.length; i++)
    {
        if (arr[i] != num)
        {
            a[j] = arr[i];
            j++;
        }
    }
    return a;
}
```

```
//--- פעולה הסופרת ומחזירה כמה מאיברי ---
//--- המערך שונים מהמספר שהתקבל ---
public static int countNums(int[]arr, int num)
{
    int count = 0;
    for (int i = 0; i < arr.length; i++)
        if (arr[i] != num)
            count++;
    return count;
}
```

C#

```
//--- פעולה המחזירה מערך חדש שבו כל האיברים ---
//--- שלא נמצאים במערך הנתון ---
public static int[] Filter (int[]arr, int num)
{
    int n = CountNums(arr, num);
    int[] a = new int[n];
    int j = 0;
    for(int i = 0; i < arr.Length; i++)
    {
        if (arr[i] != num)
        {
            a[j] = arr[i];
            j++;
        }
    }
    return a;
}
```

```
//--- פעולה הסופרת ומחזירה כמה מאיברי ---
//--- המערך שונים מהמספר שהתקבל ---
public static int CountNums(int[]arr, int num)
{
    int count = 0;
    for (int i = 0; i < arr.Length; i++)
        if (arr[i] != num)
            count++;
    return count;
}
```

עמוד 2

Java

```

public class ReportCard
{
    private String stuName; // שם התלמיד
    private Subject[] subArray; // מערך הציונים

    //--- פעולה בונה ---
    public ReportCard(String name, int num)
    {
        this.stuName = name;
        this.subArray = new Subject[num];
    }

    //--- האם התלמיד מצטיין ---
    public boolean IsExcellent()
    {
        if (this.Average() < 85)
            return false;
        boolean grd100 = false;
        int grd;
        for (int i = 0; i < subArray.length; i++)
        {
            grd = subArray[i].getGrade();
            if (grd <= 54)
                return false;
            if (grd == 100)
                grd100 = true;
        }
        return grd100; // return grd100 == true;
    }

    //--- פעולה המקבלת מערך של תעודות ומדפיסה את שמות התלמידים המצטיינים ---
    public static void PrintExcellent(ReportCard[] array)
    {
        for (int i = 0; i < array.length; i++)
        {
            if (array[i].IsExcellent())
                System.out.println(array[i].GetStuName());
        }
    }
}

```

C#

```
class ReportCard
{
    private string stuName; // שם התלמיד
    private Subject[] subArray; // מערך הציונים

    //--- פעולה בונה ---
    public ReportCard(string name, int num)
    {
        this.stuName = name;
        this.subArray = new Subject[num];
    }

    //--- האם התלמיד מצטיין ---
    public bool IsExcellent()
    {
        if (this.Average() < 85)
            return false;
        bool grd100 = false;
        int grd;
        for (int i = 0; i < subArray.Length; i++)
        {
            grd = subArray[i].getGrade();
            if (grd <= 54)
                return false;
            if (grd == 100)
                grd100 = true;
        }
        return grd100; // return grd100 == true;
    }

    //--- פעולה המקבלת מערך של תעודות ומדפיסה את שמות התלמידים המצטיינים ---
    public static void PrintExcellent(ReportCard[] array)
    {
        for (int i = 0; i < array.Length; i++)
        {
            if (array[i].IsExcellent())
                Console.WriteLine(array[i].GetStuName());
        }
    }
}
```

שאלה 3

Java

```

//--- פעולה ההופכת מחרוזת להיות מחרוזת מיוחדת ---
public static MyString Special (MyString ms)
{
    MyString str = new MyString();
    char ch;
    int n;

    while( ! ms.isEmpty())
    {
        ch = ms.firstChar();           // התו הראשון במחרוזת
        n = ms.countChar(ch);         // מספר המופעים שלו
        ms.removeChar(ch);           // מחיקתו ממחרוזת המקור

        //--- הוספת התו למחרוזת החדשה במספר המתאים של פעמים ---
        for (int i = 0; i < n; i++)
            str.appendChar(ch);
    }

    return str;
}

```

C#

```

//--- פעולה ההופכת מחרוזת להיות מחרוזת מיוחדת ---
public static MyString Special (MyString ms)
{
    MyString str = new MyString();
    char ch;
    int n;

    while( ! ms.IsEmpty())
    {
        ch = ms.FirstChar();           // התו הראשון במחרוזת
        n = ms.CountChar(ch);         // מספר המופעים שלו
        ms.RemoveChar(ch);           // מחיקתו ממחרוזת המקור

        //--- הוספת התו למחרוזת החדשה במספר המתאים של פעמים ---
        for (int i = 0; i < n; i++)
            str.AppendChar(ch);
    }

    return str;
}

```

פרק שני

שאלה 4

Java

```

//--- פעולה המקבלת רשימה של מספרים שלמים באורך זוגי ---
//--- ומחזירה רשימה כפולה כך שבראשונה מחצית המספרים הגדולים ---
//--- ובשנייה שאר המספרים הקטנים ---
public static BiList GenerateBiList (Node<Integer>lst)
{
    int k = Size(lst)/2;
    BiList bl = new BiList();
    int max;

    //--- העברת מחצית הרשימה המכילה את המספרים הגדולים ל- bl.lst1 ---
    for(int i = 0; i < k; i++)
    {
        max = MaxInList(lst);
        lst = Delete(lst, max);
        bl.addNum(max, codelist: 1);
    }

    //--- העברת שאר האיברים ל- bl.lst2 ---
    while (lst != null)
    {
        bl.addNum(lst.getValue(), codelist: 2);
        lst = lst.getNext();
    }

    return bl;
}

//--- פעולה המחזירה את האבר שהתקבל מהרשימה ---
//--- הפעולה נתונה ואין צורך לממש אותה במבחן ---
public static int Size(Node<Integer> lst)
{
    int count = 0;
    Node<Integer> pos = lst;
    while (pos != null)
    {
        count++;
        pos = pos.getNext();
    }
    return count;
}

//--- פעולה המחזירה את המספר הגדול ביותר ברשימה ---
public static int MaxInList(Node<Integer> lst)
{
    int max = lst.getValue();
    Node<Integer> pos = lst.getNext();
    while (pos != null)
    {
        if (pos.getValue() > max)
            max = pos.getValue();
        pos = pos.getNext();
    }
    return max;
}

//--- פעולה המוחקת את האיבר שהתקבל מהרשימה ---
public static Node<Integer> Delete(Node<Integer>lst, int num)
{
    if (lst.getValue() == num)
        return lst.getNext();
    Node<Integer> prev = lst;
    Node<Integer> pos = lst.getNext();
    while (pos != null)
    {
        if (pos.getValue() == num)
        {
            prev.setNext(pos.getNext());
            return lst;
        }
        else
        {
            prev = pos;
            pos = pos.getNext();
        }
    }
    return lst;
}

```

C#

```

//--- פעולה המקבלת רשימה של מספרים שלמים באורך זוגי ---
//--- ומחזירה רשימה כפולה כך שבראשונה מחצית המספרים הגדולים ---
//--- ובשנייה שאר המספרים הקטנים ---
public static Bilist GenerateBilist (Node<int>lst)
{
    int k = Size(lst)/2;
    Bilist bl = new Bilist();
    int max;

    //--- העברת מחצית הרשימה המכילה את המספרים הגדולים ל- bl.lst1 ---
    for(int i = 0; i < k; i++)
    {
        max = MaxInList(lst);
        lst = Delete(lst, max);
        bl.AddNum(max, 1);
    }

    //--- העברת שאר האיברים ל- bl.lst2 ---
    while (lst != null)
    {
        bl.AddNum(lst.GetValue(), 2);
        lst = lst.GetNext();
    }

    return bl;
}

//--- פעולה המחזירה את המספר הגדול ביותר ברשימה ---
public static int MaxInList(Node<int> lst)
{
    int max = lst.GetValue();
    Node<int> pos = lst.GetNext();
    while (pos != null)
    {
        if (pos.GetValue() > max)
            max = pos.GetValue();
        pos = pos.GetNext();
    }
    return max;
}

//--- פעולה המוחזקת את האיבר שהתקבל מהרשימה ---
//--- הפעולה נתונה ואין צורך לממש אותה במבחן ---
public static Node<int> Delete(Node<int>lst, int num)
{
    if (lst.GetValue() == num)
        return lst.GetNext();
    Node<int> prev = lst;
    Node<int> pos = lst.GetNext();
    while (pos != null)
    {
        if (pos.GetValue() == num)
        {
            prev.SetNext(pos.GetNext());
            return lst;
        }
        else
        {
            prev = pos;
            pos = pos.GetNext();
        }
    }
    return lst;
}

```

שאלה 5

Java

```
//--- פעולה המבצעת סיבובים מעגליים על רשימה ---
//--- סיבוב מעגלי: העברת האיבר האחרון לתחילת הרשימה ---
public static Node<Integer> move(Node<Integer>lst, int n)
{
    for (int i = 0; i < n; i++)
    {
        int x = removeLast(lst);
        lst = new Node<Integer>(x, lst);
        show(lst);
    }
    return lst;
}

//--- פעולה המוחקת ומחזירה את האיבר האחרון ברשימה ---
//--- הנחה: האיבר האחרון אינו האיבר היחיד ברשימה ---
public static int removeLast(Node<Integer>lst)
{
    Node<Integer> prev = null;
    Node<Integer> pos = lst;
    int x;

    while(pos.hasNext())
    {
        prev = pos;
        pos = pos.getNext();
    }
    x = pos.getValue();
    prev.setNext(pos.getNext());
    return x;
}
```

יעילות הפעולה removeLast היא $O(n)$ כאשר n מייצג את מספר האיברים ברשימה הפעולה עוברת פעם אחת על הרשימה, ומבצעת פעולות (ממשק) שיעילות כל אחת מהן היא $O(1)$

יעילות הפעולה Move היא $O(n^2)$ כאשר n מייצג את מספר ההעברות שיש לבצע (גם כן n כי ניתן לבצע העברות כמספר האיברים ברשימה, פחות, או יותר ממספר זה) הפעולה מבצעת n פעמים את הפעולה removeLast ולכן: $f(n) = n * O(n) \Leftrightarrow O(n^2)$

C#

```
//--- פעולה המבצעת סיבובים מעגליים על רשימה ---
//--- סיבוב מעגלי: העברת האיבר האחרון לתחילת הרשימה ---
public static Node<int>Move(Node<int>lst, int n)
{
    for (int i = 0; i < n; i++)
    {
        int x = RemoveLast(lst);
        lst = new Node<int>(x, lst);
        Show(lst);
    }
    return lst;
}

//--- פעולה המוזקת ומחזירה את האיבר האחרון ברשימה ---
//--- הנחה: האיבר האחרון אינו האיבר היחיד ברשימה ---
public static int RemoveLast(Node<int>lst)
{
    Node<int> prev = null;
    Node<int> pos = lst;
    int x;

    while(pos.HasNext())
    {
        prev = pos;
        pos = pos.GetNext();
    }
    x = pos.GetValue();
    prev.SetNext(pos.GetNext());
    return x;
}
```

יעילות הפעולה removeLast היא $O(n)$ כאשר n מייצג את מספר האיברים ברשימה הפעולה עוברת פעם אחת על הרשימה, ומבצעת פעולות (ממשק) שיעילות כל אחת מהן היא $O(1)$

יעילות הפעולה Move היא $O(n^2)$ כאשר n מייצג את מספר ההעברות שיש לבצע (גם כן n כי ניתן לבצע העברות כמספר האיברים ברשימה, פחות, או יותר ממספר זה)

הפעולה מבצעת n פעמים את הפעולה removeLast ולכן: $f(n) = n * O(n) \Leftrightarrow O(n^2)$

שאלה 6

	0	1	2	3	4	length
a	5	4	15	12	2	5

java: sod1 (a, 8, a.length-1)

c#: Sod1 (a, 8, a.Length-1)

x	i	i == -1	arr[i]	arr[i] == x	משפט זימון	ערך מוחזר
8	4	F	2	F	sod(a, 8, 3)	F
8	3	F	12	F	sod(a, 8, 2)	
8	2	F	15	F	sod(a, 8, 1)	
8	1	F	4	F	sod(a, 8, 0)	
8	0	F	5	F	sod(a, 8, -1)	
8	-1	T				F

א. (1) ערך מוחזר: false

א. (2) הפעולה מחזירה אמת אם x נמצא במערך ושקר אחרת

א. (3) סיבוכיות הפעולה $O(n)$ כאשר n מייצג את מספר האיברים במערך ($n == a.length$)

java: sod2 (a, 16, a.length-1)

c#: Sod2 (a, 16, a.Length-1)

x	i	i == 0	arr[i]	x - arr[i]	sod1	משפט זימון	ערך מוחזר
16	4	F	2	14	F	sod2(a, 16, 3)	
16	3	F	12	4	T		True

א. (1) ערך מוחזר: true

א. (2) הפעולה מחזירה אמת אם קיימים במערך שני מספרים שסכומם x ושקר אחרת

א. (3) סיבוכיות הפעולה $O(n^2)$ כאשר n מייצג את מספר האיברים במערך ($n == a.length$)
 הפעולה מפעילה לכל היותר n פעמים את הפעולה sod1 שיעילותה $O(n)$
 ולכן: $f(n) = n * O(n)$
 $\Rightarrow O(n^2)$

Java

```

//--- פעולה המחזירה אמת אם שני התורים בעלי אותו אורך ---
//--- ואותם איברים, גם לאחר העברת איבר מהסוף להתחלה ---
public static boolean IsSame(Queue<Integer> q1, Queue<Integer> q2)
{
    int k = size(q2);
    while(k > 0)
    {
        if (IsIdentical(q1, q2))
            return true;
        rotate(q2);
        k--;
    }
    return false;
}

//--- פעולה המחזירה אמת אם שני התורים זהים ושקר אחרת ---
public static boolean IsIdentical(Queue<Integer>q1, Queue<Integer>q2)
{
    Queue<Integer> qTemp1 = new Queue<Integer>();
    Queue<Integer> qTemp2 = new Queue<Integer>();
    boolean same = true;

    //--- העברת האיברים הזחים לתורי עזר ---
    while (! q1.isEmpty() && !q2.isEmpty() && same)
    {
        if (q1.head() != q2.head())
            same = false;
        qTemp1.insert(q1.remove());
        qTemp2.insert(q2.remove());
    }

    //--- אם אחד התורים לא ריק - האורכים שונים ---
    if (!q1.isEmpty() || !q2.isEmpty())
        same = false;

    //--- העברת שאר האיברים לתורי העזר ---
    while (!q1.isEmpty())
        qTemp1.insert(q1.remove());
    while (!q2.isEmpty())
        qTemp2.insert(q2.remove());

    //--- החזרת האיברים לתורים המקוריים ---
    while (!qTemp1.isEmpty())
        q1.insert(qTemp1.remove());
    while (!qTemp2.isEmpty())
        q2.insert(qTemp2.remove());

    return same;
}

//--- פעולה המעבירה את האיבר שבראש התור לסופו ---
public static void rotate(Queue<Integer>que)
{
    if (!que.isEmpty())
        que.insert(que.remove());
}

```

לולאה I: העברת כל האיברים הזחים.

הלולאה תסתיים כאשר:

* נמצא איבר שונה

* אחד התורים קצר מהאחר

במקרה כזה תבצע לולאה III+II

אם שני התורים בעלי אותו אורך וכל

איבריהם זהים, לא תבצע לולאה III+II

C#

```

//--- פעולה המחזירה אמת אם שני התורים בעלי אותו אורך ---
//--- ואותם איברים, גם לאחר העברת איבר מהסוף להתחלה ---
public static bool IsSame(Queue<int> q1, Queue<int> q2)
{
    int k = Size(q2);
    while(k > 0)
    {
        if (IsIdentical(q1, q2))
            return true;
        Rotate(q2);
        k--;
    }
    return false;
}

//--- פעולה המחזירה אמת אם שני התורים זהים ושקר אחרת ---
public static bool IsIdentical(Queue<int>q1, Queue<int>q2)
{
    Queue<int> qTemp1 = new Queue<int>();
    Queue<int> qTemp2 = new Queue<int>();
    bool isSame = true;

    //--- העברת האיברים הזהים לתורי עזר ---
    while (!q1.IsEmpty() && !q2.IsEmpty() && isSame)
    {
        if (q1.Head() != q2.Head())
            isSame = false;
        qTemp1.Insert(q1.Remove());
        qTemp2.Insert(q2.Remove());
    }
    //--- אם אחד התורים לא ריק - האורכים שונים ---
    if (!q1.IsEmpty() || !q2.IsEmpty())
        isSame = false;

    //--- העברת שאר האיברים לתורי העזר ---
    while (!q1.IsEmpty())
        qTemp1.Insert(q1.Remove());
    while (!q2.IsEmpty())
        qTemp2.Insert(q2.Remove());

    //--- החזרת האיברים לתורים המקוריים ---
    while (!qTemp1.IsEmpty())
        q1.Insert(qTemp1.Remove());
    while (!qTemp2.IsEmpty())
        q2.Insert(qTemp2.Remove());

    return isSame;
}

//--- פעולה המעבירה את האיבר שבראש התור לסופו ---
public static void Rotate (Queue<int>que)
{
    if (!que.IsEmpty())
        que.Insert(que.Remove());
}

```

לולאה I: העברת כל האיברים הזהים.

הלולאה תסתיים כאשר:

* נמצא איבר שונה

* אחד התורים קצר מהאחר

במקרה כזה תבצע לולאה III+II

אם שני התורים בעלי אותו אורך וכל איביהם זהים, לא תבצע לולאה III+II

כרף פליסי

לפניך שאלות מ-4 מסלולים שונים: מערכות מחשב ואסמבלי (שאלות 8-9), מבוא לחקר ביצועים (שאלות 10-11), מודלים חישוביים (שאלות 12-13), תכנות מונחה עצמים בשפת Java (שאלות 14-15), תכנות מונחה עצמים בשפת C# (שאלות 16-17).

מערכות מחשב ואסמבלי

פתרון לפרק זה נכתב ע"י: רונית (מרציאנו) גל-אור

טאלף 8

סעיף א

(1)

AX		BX		CX		DX	
AH	AL	BH	BL	CH	CL	DH	DL
	06h	00h	5Ch				8Dh
		00h	5Bh		3Fh		
	05h				40h		
		00h	5Ah		0ABh		
	04h				0ACh		
		00h	59h		71h		
					72h		
	03h	00h	58h		0C5h		
					0C6h		
	02h	00h	57h		0E2h		
					0E3h		
	01h	00h	56h		59h		
					5Ah		
	00h	00h	55h				
		00h	56h				8Eh

address	56h	57h	58h	59h	5Ah	5Bh	5Ch
value	59h	0E2h	0C5h	71h	0ABh	3Fh	8Dh
	8Eh	5Ah	0E3h	0C6h	72h	0ACh	40h

(2) הקטע מבצע הזזה ציקלית 1 ימינה של כל הערכים, תוך כדי הוספת 1 לכל הערכים.

טעיף ב

REZ DW ?

```
POP AX
POP BX
CMP AX,0
JG GO1
NEG AX
GO1:
CMP BX,0
JG GO2
NEG BX
GO2:
CMP AX,BX
JG GO3
MOV REZ,BX
JMP SOF
GO3:
MOV REZ, AX
SOF:
```

שאלה 9

```
ARR DB 10 DUP(0)
VAL DB ?
IND DB ?
```

```
MOV BX,08H
CMP IND,9 ; אם האינדקס הוא אחרון יש לטפל בו בנפרד
JNE AGAIN
MOV AL,VAL
MOV [BX+1],AL
JMP SOF
```

```
AGAIN:
MOV AL,[BX]
MOV [BX+1],AL
DEC BX
CMP BL,IND
JGE AGAIN
INC BX
MOV AL,VAL
MOV [BX],AL
SOF:
```

פתרון נוסף לשאלה 9
נכתב ע"י אריז אביב

```
MOV AH,VAL
```

```
MOV BL,IND
MOV BH,BH
```

```
AGAIN:
MOV AL,[BX]
MOV [BX],AH
```

```
MOV AH,AL
INC BX
CMP BX,09
JLE AGAIN
```

מבוא לחקר ביצועים

עלף 10

עלף 11

מודלים חישוביים

נכתב ע"י: רחל לודמר

שאלה 12

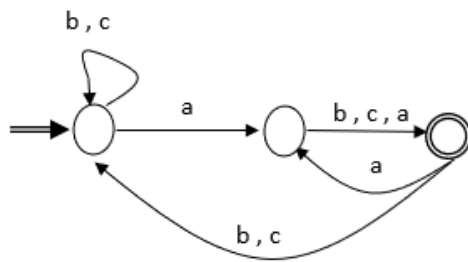
א.1. $cbccbcaac \in L$

א.2. $bbc \notin L$

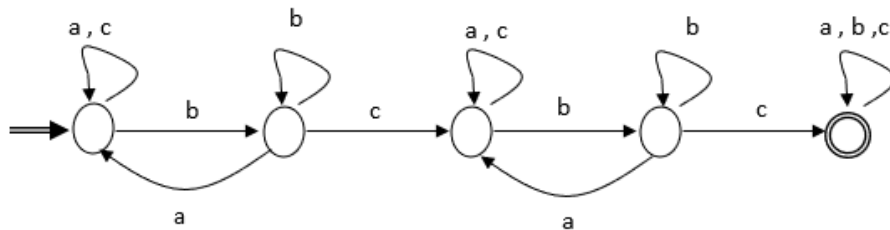
ב.

בנה אוטומטים סופיים דטרמיניסטיים לשפות הבאות:

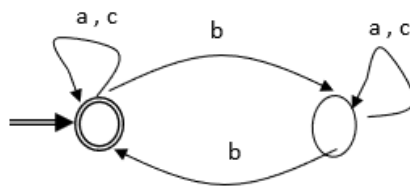
$L_1 = \{a \text{ האות שלפני האחרונה היא } a\}$



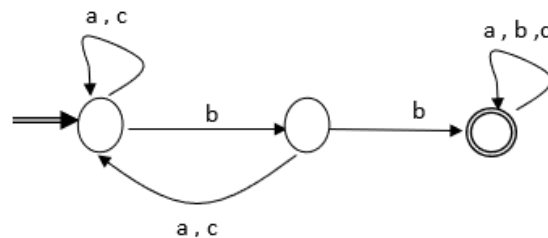
$L_2 = \{bc \text{ לפחות פעמיים}\}$



$L_3 = \{\text{מספר } b \text{ זוגי}\}$



$L_3 = \{bb \text{ מכיל}\}$



לשפות L_1, L_2, L_3 בנינו אוטומט סופי דטרמיניסטי לכן הן רגולריות.

מתכונות סגירות ש שפות רגולריות, משלים של שפה רגולרית הוא רגולרי, וחיתוך של שפות רגולריות הוא רגולרי.

לכן השפה L רגולרית

מכאן $L = L_1 \cap L_2 \cap \overline{L_3}$ היא רגולרית.

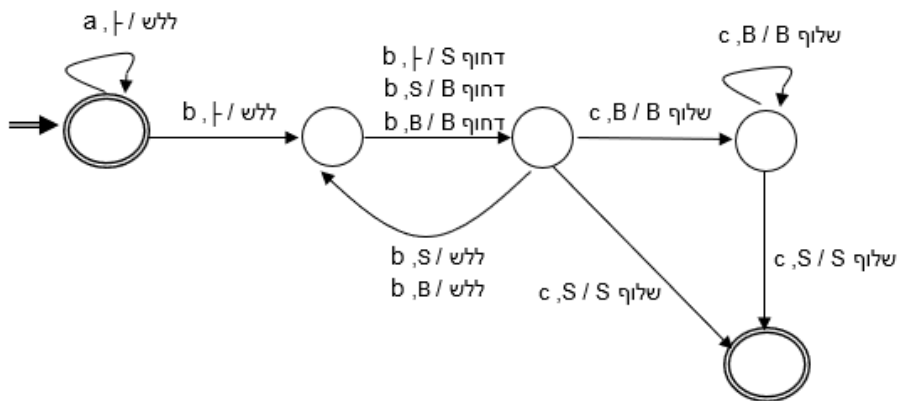
שאלה 13

$$L_1 = \{a^n b^{2k} c^k \mid n, k \geq 0\}$$

$$L_2 = \{a^n b^m c^k \mid n, m, k > 0, n \% 2 = m \% 2 = k \% 2\}$$

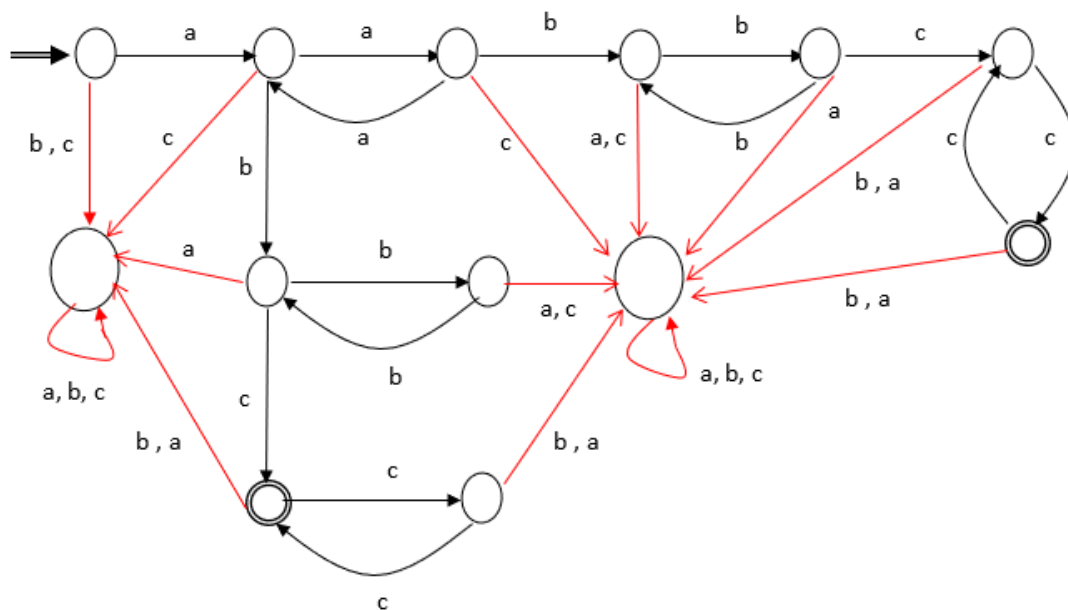
א.

השפה L_1 היא לא רגולרית, קיימת תלות של מניה בין רצפי b לבין רצפי c. לכן נבנה אוטומט מחסנית.



ב.

השפה L_2 היא רגולרית. כל הרצפים הם זוגיים או אי זוגיים. לכן נבנה אס"ד מתאים.



תכנות מונחה עצמים בשפת Java

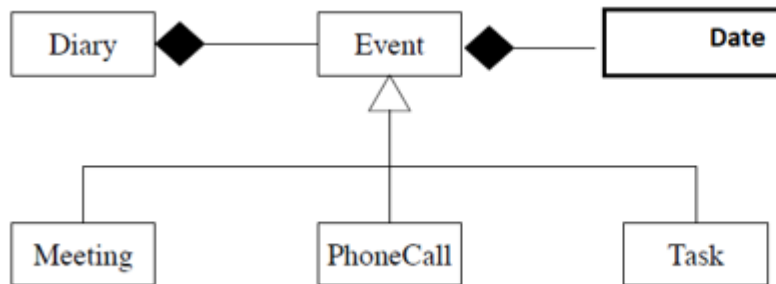
עלף 14

עלף 15

תכנות מונחה עצמים בשפת C#

נכתב ע"י: דיתה אוהב ציון

שאלה 16



א.

```

public class Event
{
    private Date date;
    private int hour;
}

```

```

//הנחה: יש בדיוק 100 שיחות טלפון ביום
public PhoneCall[] AllCalls(Date date)
{
    PhoneCall[] arr = new PhoneCall[100];
    int index = 0;
    foreach( Event e in arr)
    {
        if(e is PhoneCall)
        {
            if(e.GetDate().Same(date) )
            {
                string p = ((PhoneCall)e).GetPhone();
                string n = ((PhoneCall)e).GetName();
                arr[index++] = new PhoneCall(date, e.GetHour(),p , n);
            }
        }
    }
    return arr;
}

```

ב.

ג.

במחלקה Event

```
public virtual bool Match(string name) { return false; }
```

במחלקה Meeting

```
public override bool Match(string name)
{
    for (int i = 0; i < arrNames.Length; i++)
    {
        if (arrNames[i] == name) return true;
    }
    return false;
}
```

במחלקה PhoneCall

```
public override bool Match(string name)
{
    return this.name == name;
}
```

עלף 17

1	A a = new A(); a.Foo(2);	ctor A A Foo int2
2	A a = new A(); ((B)a).Foo(3);	שורה 1 תקין לא תקין. שגיאת זמן ריצה בהמרה. אי אפשר להמיר לעצם לא מוכר. הפניה מטיפוס A המחזיקה עצם מטיפוס A לא מכירה את B.
3	A a = new A(); B b = a; b.Foo(2);	שגיאת קומפילציה בשורה 2. בן לא יכול להחזיק אבא.
4	A x = new B(); x.Foo(2);	ctor A ctor B B Foo int2
5	A x = new B(); x.Bar();	שגיאת קומפילציה בשורה 2. הפניה מטיפוס A מפעילה את פעולה Bar שלה, וחסר פרמטר הנדרש בחתימת הפעולה.
6	A a = new A(); a.Bar(3);	ctor A A Bar3 A Foo int3

7	<code>A a = new A(); a.Bar();</code>	שגיאת קומפילציה. חסר פרמטר לפעולה Bar
8	<code>B b = new B(); b.Bar();</code>	ctor A ctor B B Bar B Foo int2
9	<code>B b = new B(); b.Bar(3);</code>	ctor A ctor B A Bar3 B Foo int3
10	<code>B b = new B(); b.Foo(2); b.Foo(2.0);</code>	ctor A ctor B B Foo int2 A Foo double2
11	<code>A a = new A(); a.Foo(2); a.Foo(2.0);</code>	ctor A A Foo int2 A Foo double2
12	<code>B b = new A(); b.Another(2);</code>	שגיאת קומפילציה בשורה הראשונה. הפניה מטיפוס הבן לא יכולה להחזיק עצם מטיפוס האב.
13	<code>A a = new A(); a.Another(2);</code>	שגיאת קומפילציה. הפעולה אינה מוגדרת במחלקה A
14	<code>B x = new B(); x.Another(2);</code>	ctor A ctor B B Another2 A Foo int2 A Foo double4
15	<code>A x = new B(); x.Another(2);</code>	שגיאת קומפילציה. הפניה מטיפוס A אינה מכירה פעולות מהמחלקות היורשות. (חסרה המרה מפורשת - ((B)x).Another(2);

בהצלחה !