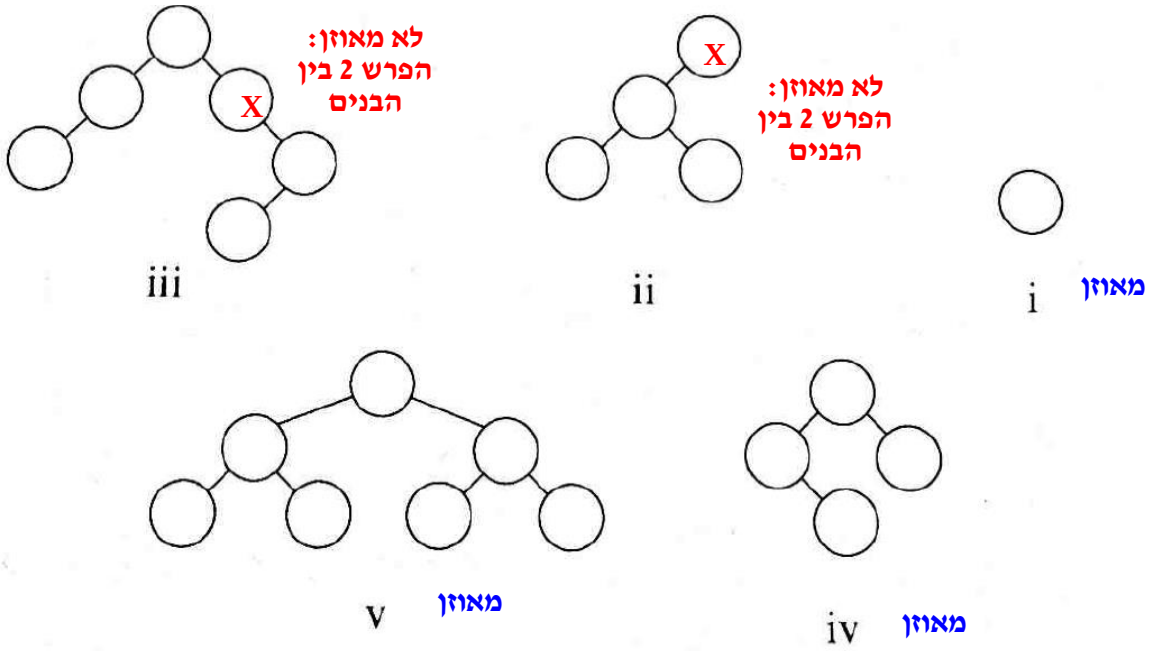
מדעי המחשב ב'

פתרון בחינת הבגרות

פרק א - עיצוב תכנה

שאלה 1

א.



Java:

```
//--- פעולה המחזירה אמת אם העץ מאוזן ושקר אחרת ---
public static boolean isBalanced (BinTreeNode<Integer>bt)
{
    if (bt == null)
        return true;
    if (Math.abs(height(bt.getLeft()) - height(bt.getRight())) > 1)
        return false;
    return isBalanced (bt.getLeft()) && isBalanced (bt.getRight());
}

//--- פעולה המחזירה את גובה העץ ---
public static int height (BinTreeNode<Integer>bt)
{
    if (bt == null)
        return -1;
    return 1 + Math.max(height(bt.getLeft()), height(bt.getRight()));
}
```

פתרון #C - נכתב ע"י **ראמי ג'באלי**

```
// פעולה מחזירה גובה עץ
public static int Height(BinTreeNode<int> t)
{
    if (t == null)
        return -1;
    return Math.Max(Height(t.GetLeft()), Height(t.GetRight())) + 1;
}

//פעולה מחזירה אמת אם העץ הוא עלה אחרת מחזירה שקר
public static bool IsLeaf(BinTreeNode<int> t)
{
    return (t.GetLeft() == null && t.GetRight() == null);
}

//פעולה מחזירה אמת אם העץ מאוזן אחרת מחזירה שקר
public static bool IsMozan(BinTreeNode<int> t)
{
    if (t == null || IsLeaf(t))
        return true;
    if (Math.Abs(Height(t.GetLeft())-Height(t.GetRight()))>1)
        return false;
    return IsMozan(t.GetLeft()) && IsMozan(t.GetRight());
}
```

פתרון #C **בתכנית החדשה** - נכתב ע"י **דיתה אוהב ציון**

```
/// פעולת עזר המחזירה את גובה העץ
/// <param name="t">העץ לבדיקה</param>
/// <returns>מספר שלם</returns>
public static int High(BinNode<int> t)
{
   if (t == null)    return -1;
   return 1+ Math.Max(High(t.GetLeft()),High(t.GetRight()));

}

/// פעולה המקבלת עץ ומחזירה אמת אם הוא מאוזן, אחרת תחזיר שקר
/// <param name="t">העץ לבדיקה</param>
/// <returns>אמת / שקר</returns>
public static bool IsBalance(BinNode<int> t)
{
   if (t == null)  return true;
   if (Math.Abs(High(t.GetLeft())-High(t.GetRight()))>1)
      return false;
   return IsBalance(t.GetLeft()) && IsBalance(t.GetRight());

}
```

שאלה 2 Java:

```
public class Collec
{
    private List<Integer>lst;
    private int max, min;

    //--- ב. פעולה בונה ללא פרמטרים ---
    public Collec ()
    {
        this.lst = new List<Integer>(); //-- אפשר גם תור או מחסנית ---
        this.max = -1;
        this.min = -1;
    }

    //--- פעולה בונה עם פרמטרים ---
    public Collec (int n)
    {
        this.lst = new List<Integer>();
        this.max = n;
        this.min = n;
        this.lst.insert(null, n);
    }

    //--- ג. פעולה המוספיה איבר לאוסף ---
    public boolean add (int num)
    {
        if (num > max)
        {
            this.lst.insert(null, num);
            this.max = num;
            if (this.min == -1)
                this.min = num;
            return true;
        }
        return false;
    }

    //--- ד. פעולה המחזירה את האיבר הקטן ביותר באוסף ---
    public int small ()
    {
        return this.min;
    }

    //--- ה. פעולה המחזירה את המספר הקטן ביותר מבין שני האוספים ---
    public int smallest (Collec c)
    {
        return Math.min(this.small(), c.small());
    }
```

פתרון **Java** - נכתב ע"י **יבגני קנל**

```
import unit4.collectionsLib.*;
// סעיף א. כותרת מחלקה ותכונות שלה
public class Collec
{
        private int min;          // מספר הקטן באוסף
        private Stack<Integer> data;    // נתונים באוסף

        // סעיף ב. פעולה בונה ללא פרמטרים
        public Collec()  {
                this.min=-1;
                this.data=new Stack<Integer>();
        }

        // סעיף ב. פעולה בונה המקבלת מספר שלם גדול מ-0

        public Collec(int n)  {
                this.min=n;
                this.data=new Stack<Integer>();
                this.data.push(n);
        }

        // סעיף ג. פעולה המקבלת ערך גדול מ-0 ומוסיפה אותו לאוסף
        public boolean add(int n)  {
                if(this.min==-1)   {
                        this.data.push(n);
                        this.min=n;
                        return true;
                }

                if( n>=this.data.top()) {
                        this.data.push(n);
                        return true;
                }
                return false;
        }

        // סעיף ד. פעולה המחזירה ערך קטן ביותר באוסף
        public int small()  {
                return this.min;
        }

        // סעיף ה. פעולה המחזירה ערך הקטן ביותר בין שני אוספים. הנכה: שני האוספים אינם ריקים
        public int smallest(Collec c)  {
                return Math.min(this.small(), c.small());
        }

         public String toString()  {
                return this.min+"  -->"+this.data;
        }
}
```

פתרון #C - נכתב ע"י **ראמי ג'באלי**

```
public class Collec
{
    private List<int> list;

    public Collec()
    {
        list = new List<int>();
    }

    public Collec(int n)
    {
        list = new List<int>();
        list.Insert(null, n);
    }

    public int GetMax()
    {
       int max=0;
       Node<int> pos = list.GetFirst();
       while (pos != null)
       {
           if (pos.GetInfo() > max)
               max = pos.GetInfo();
           pos = pos.GetNext();
       }
       return max;
    }

    public bool Add(int x)
    {
        if (x > GetMax())
        {
            list.Insert(null, x);
            return true;
        }
        return false;
    }
    public int Samll()
    {
        Node<int> pos = list.GetFirst();
        int min = pos.GetInfo();
        pos = pos.GetNext();
        while (pos != null)
        {
            if (pos.GetInfo() < min)
                min = pos.GetInfo();
            pos = pos.GetNext();
        }
        return min;
    }

    public int Smallest(Collec c)
    {
        return Math.Min(Samll(), c.Samll());
    }
```

פתרון #C בתכנית החדשה - נכתב ע"י **דיתה אוהב ציון**

```
public  class Collec
{
        private  Node<int> first;  //  מאפיין המחלקה – הפניה לשרשרת חוליות
        private int max;                //האיבר הגדול ביותר
```

סעיף א.

סעיף ב.

```
        ///     בנאי היוצר אוסף ריק
         public Collec()
        {
                this.first = null;
                this.max = 0;
        }

        ///  בנאי המוסיף איבר לאוסף
        ///   הנחה- המספר גדול מ-0
        /// <param name="n">מספר שלם גדול מ- 0 - המספר להוספה  </param>
        public Collec(int n)
        {
                if (this.first==null)
                        this.max = n;        //  האיבר הגדול הוא הראשון
                this.first = new Node<int>(n);
        }
```

סעיף ג.

```
        /// פעולה המוסיפה מספר לאוסף בתנאי שאין גדול ממנו באוסף
        ///   הנחה- המספר גדול מ-0
        /// <param name="n">מספר שלם גדול מ- 0 - המספר להוספה</param>
        /// <returns>.אמת/שקר  אם הוסף האיבר</returns>
        public bool Add(int n)
        {
                if  (n > max )
                        return false;
                else
                {
                        if  (this.first == null)
                                this.first = new Node<int>(n);
                        else
                                this.first.SetNext(new Node<int>(n, first.GetNext()));
                }
                return true;
        }
```

סעיף ד.

```
        /// פעולה המחזירה מהו המספר הקטן ביותר באוסף ///
        /// <returns> מספר שלם</returns>
        public int Small()
        {
                Node<int> pos = this.first;
                int min = int.MaxValue;
                while (pos != null)
                {
                        if (pos.GetValue() < min)
                                min = pos.GetValue();
                        pos = pos.GetNext();
                }
                return min;
        }
```

סעיף ה.

```
        /// פעולה המקבלת אוסף ומחזירה מהו המספר הקטן בין שני האוספים ///
        /// <param name="c"></param>
        /// <returns>מספר שלם- המספר הקטן ביותר באוספים</returns>
        public int Smallest(Collec c)
        {
                return Math.Min(Small(), c.Small());
        }
}
```

שאלה 3 Java :

בהנחה שבמחלקה School קיימת הפעולה הבאה: (שהיא פעולה הגיונית יותר מפעולה המחזירה הפניה למערך)

```
//--- פעולה המחזירה הפנייה לתחילת הרשימה השכבתית ---
public List<Student> getClassList(int level)
{
    return this.ar[level];
}
```

אתחול מערך החודשים:

```
//--- פעולה המחזירה מערך בגודל 12 שכל איבר בו ---
//--- הוא הפניה לרשימת תלמידים שנולדו בחודש זה ---
public static List<Student>[] getBirthDayArr()
{
    List<Student>[] arr = new List[12];
    for (int i = 0 ; i < arr.length ; i++)
        arr[i] = new List<Student>();
    return arr;
}
```

הכנסת התלמידים למערך החודשים לפי חודש הלידה (בהיסט של 1):

```
//--- פעולה המקבלת עצם מטיפוס בי"ס ---
//--- ומחזירה מערך של 12 חודשים, כך ---
//--- שכל איבר במערך הינו רשימת התלמידים ---
//--- שנולדו בחודש זה (בהיסט של 1) ---
public static List<Student>[] getBirthDayList (School sc)
{
    List<Student>[] arr = getBirthDayArr();
    for (int i = 0 ; i < 6 ; i++)
    {
        List<Student> lst = sc.getClassList(i);
        Node<Student> pos = lst.getFirst();
        while (pos != null)
        {
            Student st = pos.getInfo();
            int month = st.getBirthDay().getMonth();
            arr[month-1].insert(null, st);

            pos = pos.getNext();
        }
    }
    return arr;
}
```

פתרון #C - נכתב ע"י **ראמי ג'באלי**

במחלקה School קיימת הפעולה הבאה, המחזירה הפניה למערך ביה"ס:

```
public List<Student>[] GetAr()
{

    return ar;
}
```

הכנסת התלמידים למערך החודשים לפי חודש הלידה:

```
//3 שאלה
//פעולה מחזירה  מערך בגוודל 12 כל תא במערך מייצג חודש בשנה//
public static List<Student>[] Build(School s)
{
    List<Student>[] arr = new List<Student>[12];
    for (int i = 0; i < arr.Length; i++)
        arr[i] = new List<Student>();

    for (int i = 0; i < 6; i++)
    {
        Node<Student> pos = s.GetAr()[i].GetFirst();
        while (pos != null)
        {
            int month = pos.GetInfo().GetBirthDay().GetMonth();
            arr[month-1].Insert(null, pos.GetInfo());
            pos = pos.GetNext();
        }
    }
    return arr;
}
```

פתרון **#C בתכנית החדשה** - נכתב ע"י **דיתה אוהב ציון**

```
public static  Node<Student>[] AllBirthDay(School s)
{
    int i, m;
    Node<Student>[] all = new Node<Student>[12];//מערך התוצאות
    for ( i = 0; i < all.Length; i++)
        all[i] = null;
    Node<Student>[] st = s.GetArr();  //קבלת מערך תלמידי בית הספר
    for (i = 0; i < st.Length; i++)  //  סריקת המערך
    {
        if (st[i] != null)
        {
            Node<Student> pos = st[i];
            while (pos != null)          // סריקת שרשרת החוליות
            {
                m = pos.GetValue().GetStMonth();  //  חודש הלידה של התלמיד
                if (all[m-1] == null)
                    all[m-1] = new Node<Student>(pos.GetValue());//  הוספת התלמיד
                else
                    all[m-1].SetNext(new Node<Student>(pos.GetValuue(), all[m-1].GetNext()));
                pos = pos.GetNext();

            }
        }

    }
    return all;
}
```

שאלה 4

א. lst: [-2 , -9087 , 16 , -43 , 5]

ערך מוחזר	x ≥ 0	lst	x	רשימה ריקה?
1 + _2_ = 3	לא	[-2, -9087,16,-43,5]	-2	לא
1 + _1_ = 2	לא	[-9087,16,-43,5]	-9087	לא
__1__	כן	[16,-43,5]	16	לא
1 + _0__ = 1	לא	[-43,5]	-43	לא
_0_	כן	[5]	5	לא
		[ ]		כן

(1) ערך מוחזר: 3

(2) הפעולה סופרת ומחזירה את מספר האיברים השליליים ברשימה

(3) (i) הפעולה תספור את רצף האיברים השלילי שבתחילת הרשימה, עד האיבר הלא שלילי הראשון.
(אם בתחילת הרשימה יהיה איבר חיובי, תחזיר הרשימה 0)

(ii) בסיום הפעולה תכיל הרשימה את כל האיברים שהיו בה ללא הרצף ההתחלתי של האיברים השליליים וללא האיבר החיובי הראשון.
בסיום הפעולה תכיל הרשימה הנתונה את האיברים הבאים: [-43 , 5]

ב. s: [12 , 4 , 33 , 6 , 30 , 0]

ערך מוחזר	x % 6 == 0	x	המחסנית ריקה?	s
12 + _36_ = 48	כן	12	לא	[12,4,33,6,30,0]
_36_	לא	4	לא	[4,33,6,30,0]
_36_	לא	33	לא	[33,6,30,0]
6 + _30_ = 36	כן	6	לא	[6,30,0]
30 + _0_ = 30	כן	30	לא	[30,0]
0 + _0_ = 0	כן	0	לא	[0]
			כן	[ ]

(1) ערך מוחזר: 48 (סכום המספרים שבמחסנית, המתחלקים ב- 6)

(2) מצב המחסניות והתורים לפני זימון sod מתוך main:

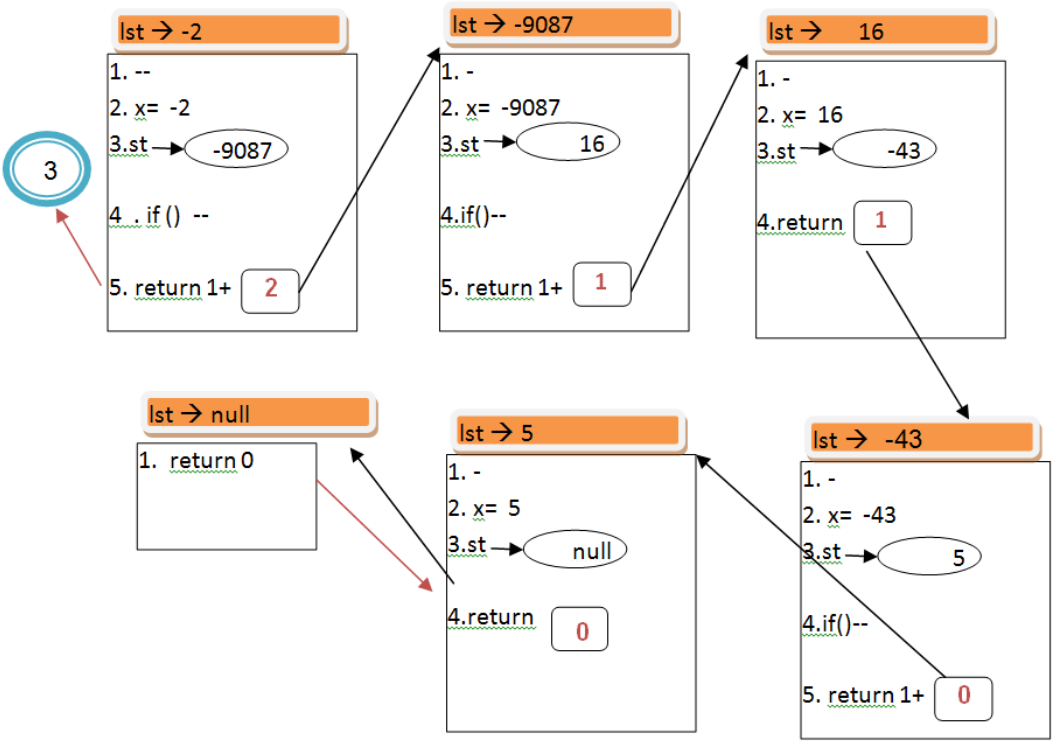
s1 : [12 , 4 , 33 , 6 , 30 , 0] s2: [1 , 36 , 23]

que: [[s1] , [s2]] qr: [ ] בזימון לפעולה מכילים התורים:

qq	qm	x
[[12,4,33,6,30,0] , [1,36,23]]	[48]	48
[[1,36,23]]	[48 , 36]	36
[ ]		

פתרון לפי **התכנית החדשה** - נכתב ע"י **דיתה אוהב ציון**

<u>סעיף א</u> –(1)מעקב (בפעולה 5 שורות הממוספרות 1-5)



הפעולה תחזיר 3. הרשימה לא תשתנה בעקבות הפעולה .

(2) הפעולה מחזירה כמה מספרים הקטנים מ-0 יש ברשימה

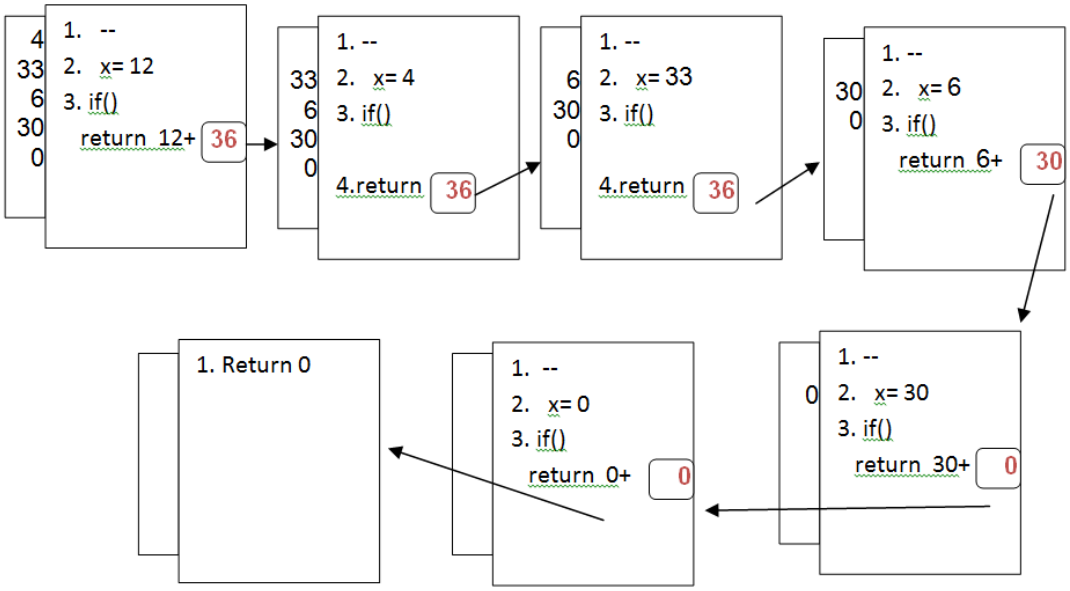
(3) א. אחרי שינוי הפעולה תעצור כשתגיע לחוליה הראשונה שערכה חיובי (16) ותחזיר 2 .

כמה מספרים הקטנים מ-0 נמצאים ברשימה לפני המספר הראשון הגדול או שווה ל-0 .

ב. הרשימה לא תשתנה אחרי ביצוע הפעולה.

סעיף ב.

(1) הפעולה מחזירה 48 - סכום המספרים במחסנית המתחלקים ב-6 ללא שארית

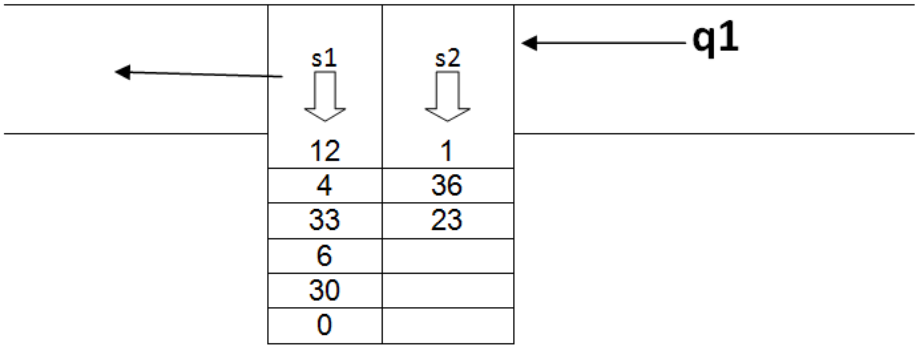


(2)

התורים הנבנים בתכנית הראשית :

qr הוא תור של מספרים המתחיל כתור ריק .

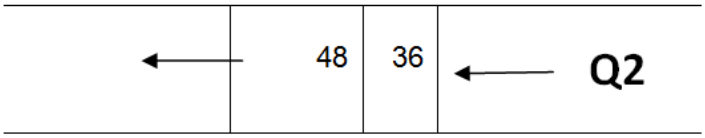
q1 הוא תור שכל איבר בו היא מחסנית של מספרים



לאחר ההרצה של הפעולה :

q1 – הוא תור ריק. שתי המחסניות הוצאו ממנו.

q2 -מכיל 2 מספרים - סכום המספרים במחסנית המתחלקים ב-6 ללא שארית – משתי המחסניות



פרק ב'

מערכות מחשב ואסמבלר

הפתרון לפרק זה נכתב ע"י: **רונית מרציאנו**

תרגיל 5

א. לפניך קטע באסמבלר

```
        MOV SI,200H
        MOV AL,[SI[
        MOV BL,0
        MOV CL,8
AA:     ROR AL,1
        JNC BB
        ADD BL,1
BB:     DEC CL
        JNZ AA
        MOV SI,200H
        MOV [SI+1],BL
```

1. טבלת מעקב:

AX		BX		CX		SI
AH	AL	BH	BL	CH	CL	0200H
	10100001B		00H		08H	
	11010000B		01H		07H	
	01101000B				06H	
	00110100B				05H	
	00011010B				04H	
	00001101B				03H	
	10000110B		02H		02H	
	01000011B				01H	
	10100001B		03H		00H	

כתובת בזיכרון	0201H	0200H
תוכן	03H	10100001B

2. הקטע סופר כמה ביטים דולקים בבית שכתובתו 200H ואת מספר הביטים הדולקים שם בכתובת 201H.
3. לאחר שינוי הפקודה JNC BB לפקודה JC BB תוכן הזיכרון יהיה:

כתובת בזיכרון	0201H	0200H
תוכן	05H	10100001B

הקטע יספור כמה ביטים אינם דולקים בכתובת 200H ואת מספר הביטים שאינם דולקים ישים בכתובת 201H.

ב. לפניך קטע תוכנית באסמבלר

```
MOV   CX,8
ROR   AX,CL
```

תוכנו של אוגר AX הוא: 1A2BH

1. אוגר AX בסיום קטע התוכנית יהיה: 2B1AH

הוראה אחת המבצעת את הנדרש

```
XCHG  AH,AL
```

תרגיל 6

א. לפניך קטע תוכנית באסמבלר
לפני ביצוע הקטע תוכנו של אוגר AX **הוא** 42H

```
MOV  CL,2
PUSH AX
SHL  AX,CL
POP  BX
ADD  AX,BX
```

1. טבלת מעקב:

AX		BX		CX		
AH	AL	BH	BL		CL	
00H	42H	00H	42H		02H	
01H	08H					
01H	4AH					00H
						42H
						ראש המחסנית

2. הקטע מכפיל את תוכנו של אוגר AX **ב** – 5
ולכן אם DL=5
פקודה אחת שתבצע את אותה פעולה תהיה:

```
MUL DL
```

ב. הפקודות החסרות בקטע הן:

i. MOV CX,9H
ii. MOV CX,0AH
iii. ADD AH,DL
iv. MOV CX,BX

תרגיל 7

א.

i. **קטע ראשון אינו מבצע את הנדרש**

```
CHECK:  POP   CX
        POP   BX
        XOR   AL,AL
        ADD   BH,1
        CMP   BH,BL
        JZ    A1
        INC   AL
A1 :    PUSH  CX
        RET
```

צ.ל:
JNZ A1

טבלת מעקב:

AX		BX		CX		כתובת חוצה
AH	AL	BH	BL	CH	CL	
	00H	03H	02H	כתובת חוצה		~~03H~~
		04H				~~02H~~
						~~כתובת חוצה~~
הבעיה: הקידום ב 1 נעשה על הבית הגבוה במקום הנמוך והקידום ב 1 מתבצע כאשר הערכים לא שווים ד'ווקא במקום כשהם שווים						ראש המחסנית

ii. **קטע שני אינו מבצע את הנדרש**

```
CHECK:  POP   BX
        POP   CX
        MOV   AL,0
        DEC   CL
        SUB   CH,CL
        JNZ   A1
        INC   AL
A1 :    PUSH  BX
        RET
```

טבלת מעקב:

AX		CX		BX		כתובת חוצה
AH	AL	CH	CL	BH	BL	
	00H	03H	02H	כתובת חוצה		~~03H~~
	01H		01H			~~02H~~
						~~כתובת חוצה~~
הבעיה: הפחתה ב 1 במקום קידום של הבית הנמוך.						ראש המחסנית

הפחתה של 1 מהבית הנמוך

iii. קטע שלישי אינו מבצע את הנדרש

```
CHECK:  POP   CX
        POP   BX
        MOV   AL,0
        SUB   CL,1
        CMP   BH,BL
        JNE   A1
        MOV   AL,1
A1 :    PUSH  CX
        RET
```

טבלת מעקב:

AX		BX		CX		כתובת חוצה-1
AH	AL	BH	BL	CH	CL	
	00H	03H	02H	כתובת חוצה		~~03H~~
	01H		01H	כתובת חוצה -1		~~02H~~
						~~כתובת חוצה~~
הבעיה: כתובת החוצה נהרסה ויש השוואה בין הבית הקטן והגדול במילה ללא מניפולציה כך שאם הם שווים לפני AL יקבל 1.						ראש המחסנית

כתובת חוצה נהרסה!

iv. קטע רביעי מבצע את הנדרש!

```
CHECK:  POP   BX
        POP   CX
        XOR   AL,AL
        ADD   BH,1
        CMP   BH,BL
        JZ    A1
        INC   AL
A1 :    PUSH  CX
        RET
```

טבלת מעקב:

AX		CX		BX		כתובת חזרה
AH	AL	CH	CL	BH	BL	
	01H	03H	02H	כתובת חזרה		~~03H~~
			03H			~~02H~~
						~~כתובת חזרה~~
					מצוין!	ראש המחסנית

ב. קטע תוכנית המבצע את הקטע ב JAVA או C#

```
X       DW  ?
Y       DW  ?

        MOV   BX,X
        MOV   CX,Y
N1:     CMP   BX,0
        JLE   SOF
        CMP   CX,0
        JLE   SOF
        CMP   BX,0AH
        JLE   N2
        DEC   BX
        JMP   CONT
N2:     DEC   CX
CONT:   JMP   N1
SOF:    NOP
```

שאלה 8

```
DATA SEGMENT
    ARR DB 100 DUP(?)
    LEN= $-ARR
    X   DB ?
ENDS

STACK SEGMENT
    DW   128  DUP(0)
ENDS

CODE SEGMENT
START:
        MOV   AX, DATA
        MOV   DS, AX
        MOV   ES, AX

        MOV   DH,1;        meozan
        MOV   CX,LEN
        LEA   BX,ARR
        ADD   BX,LEN-1
        LEA   SI,ARR
        SHR   CX,1
AGAIN:  PUSH  BX ;         last cell
        PUSH  SI;          first cell
        LEA   DI,X ;       x
        PUSH  DI
        CALL  CHECK
        POP   DI
        POP   SI
        POP   BX
        JE    BYBY
        DEC   BX
        INC   SI
        LOOP  AGAIN
        JMP   SOF

BYBY:   MOV   DH,0
SOF:    NOP
        MOV   AX, 4C00H
        INT   21H

PROC  CHECK
        MOV   BP,SP
        MOV   DI,[BP+2]
        MOV   SI,[BP+4]
        MOV   BX,[BP+6]
        MOV   DL,0
        MOV   AH,[DI]
        MOV   AL,[SI]
        ADD   AL,[BX]
        CMP   AL,AH
        JNE   CONT
        INC   DL
  CONT: RET
ENDS

END START
```

פרק ב'

מבוא לחקר ביצועים

שאלה 9

שאלה 10

שאלה 11

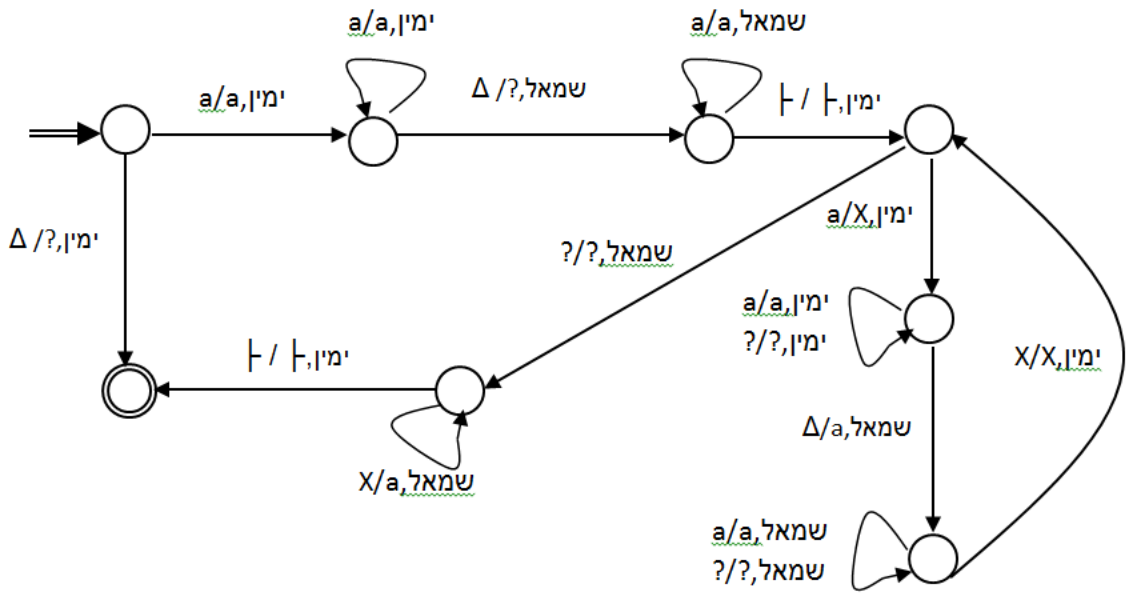
שאלה 12

פרק ב'

מודלים חישוביים

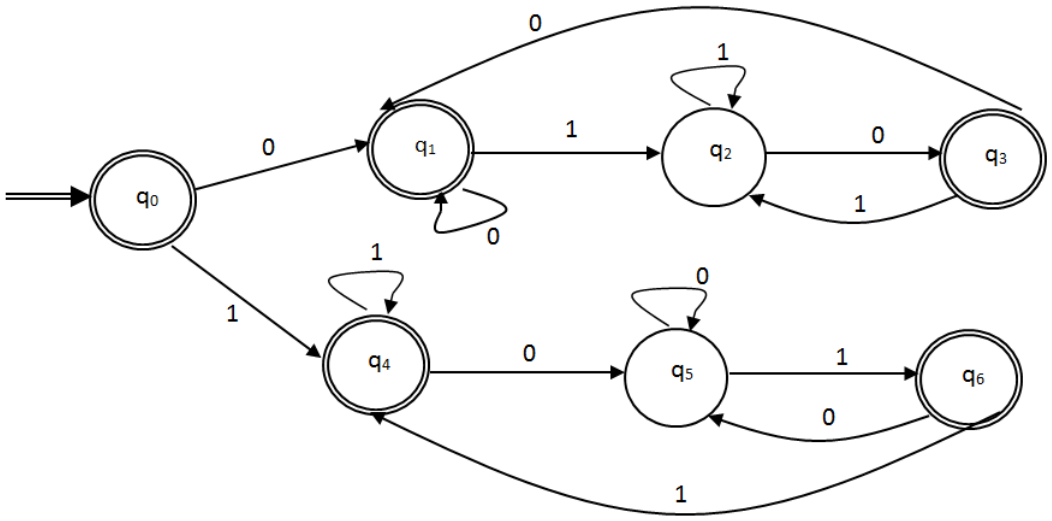
הפתרון לפרק זה נכתב ע"י **רחל לודמר**.

תרגיל 13



שאלה 14

א.



ב.

(1) דוגמא ל- L_1,L_2 רגולריות , L_3 לא רגולריות כך ש: $L_3 \cup L_2 = L_1$

$$L_1 = \{a^n b^m \mid n,m \geq 0\}$$
$$L_2 = \{a^n b^m \mid n,m \geq 0\}$$
$$L_3 = \{a^n b^m \mid n,m \geq 0, n \neq m\}$$
$$L_3 \cup L_2 = \{a^n b^m \mid n,m \geq 0, n \neq m\} \cup \{a^n b^m \mid n,m \geq 0\}$$
$$= \{a^n b^m \mid n,m \geq 0\} = L_1$$

(2) דוגמא ל- L_1,L_2 רגולריות , L_3 לא רגולריות כך ש: $L_3 \cap L_2 = L_1$

$$L_1 = \{c^n \mid n \geq 0\}$$
$$L_2 = \{b^n \mid n \geq 0\}$$
$$L_3 = \{a^n b^n c^m \mid n,m \geq 0\}$$
$$L_3 \cap L_2 = \{a^n b^n c^m \mid n,m \geq 0\} \cap \{b^n \mid n \geq 0\}$$
$$= \{c^m \mid m \geq 0\} = L_1$$

פתרון נוסף לשאלה 14, סעיף ב' - נכתב ע"י **יבגני קנל**

(1) $L_1=L_2=\sum^*$- שפות כל המילים מעל { a,b } – שפות רגולריות
$L_3=\{a^nb^n \mid n>=0\}$ – שפה לא רגולרית
$L_2 \cup L_3 = L_1$

(2) $L_1=L_2=\varnothing$ - שפות ריקות – שפות רגולריות
$L_3=\{a^nb^n \mid n>=0\}$ – שפה לא רגולרית
$L_2 \cap L_3 = L_1$

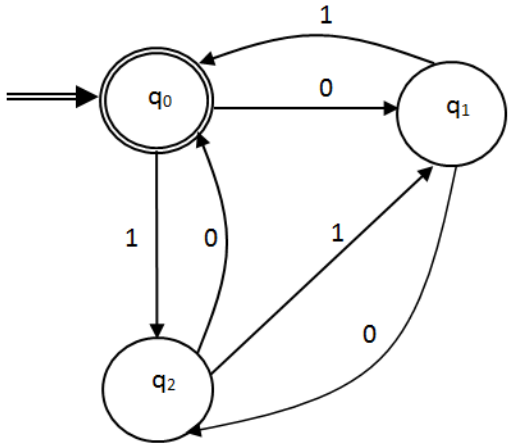
אופציה אחרת:
$L_1=\varnothing$ - שפה ריקות – שפה רגולריות
$L_2=\{a\}$ – שפה רגולרית הכוללת מילה אחת בלבד
$L_3=\{a^nb^n \mid n>=0\}$ – שפה לא רגולרית
$L_2 \cap L_3 = L_1$

שאלה 15

א השפה L הנתונה היא:

$$L = \{w \mid |0_w| \%3 = |1_w| \%3,\ w \in \{0,1\}\}$$

השלמת האוטומט:



ב. נתונות השפות:

$$L_1 = \{0^n \cdot x \mid x \in \{0,1\}, 1 \le |x| \le n\}$$
$$L_2 = \{0^m \mid m \ge 1\}$$
$$w_1 = 0^i \quad , w_2 = 0^j \ , j < i,\ w_1, w_2 \in L_2$$

המילה שנבחר: $w = 1^{j+1} \ \ j \ge 1$

$$w_1 \cdot w = 0^i 1^{j+1} \in L_1 \ | i, j \ge 1, j < i \ \Rightarrow j+1 \le i$$
$$w_2 \cdot w = 0^j 1^{j+1} \notin L_1 \mid j \ge 1 \quad \Rightarrow j+1 > j$$

ג. נתונות השפות:

$$L_1 = \{a^n b^m c^{n+m} \mid n \ge 1, m \ge 1\}$$
$$L_2 = \{a^s \mid s \ge 1\}$$
$$w_1 = a^i \quad , w_2 = a^j \ , j \ne i,\ w_1, w_2 \in L_2$$

המילה שנבחר: $w = b^j c^{i+j} \mid i, j \ge 1$

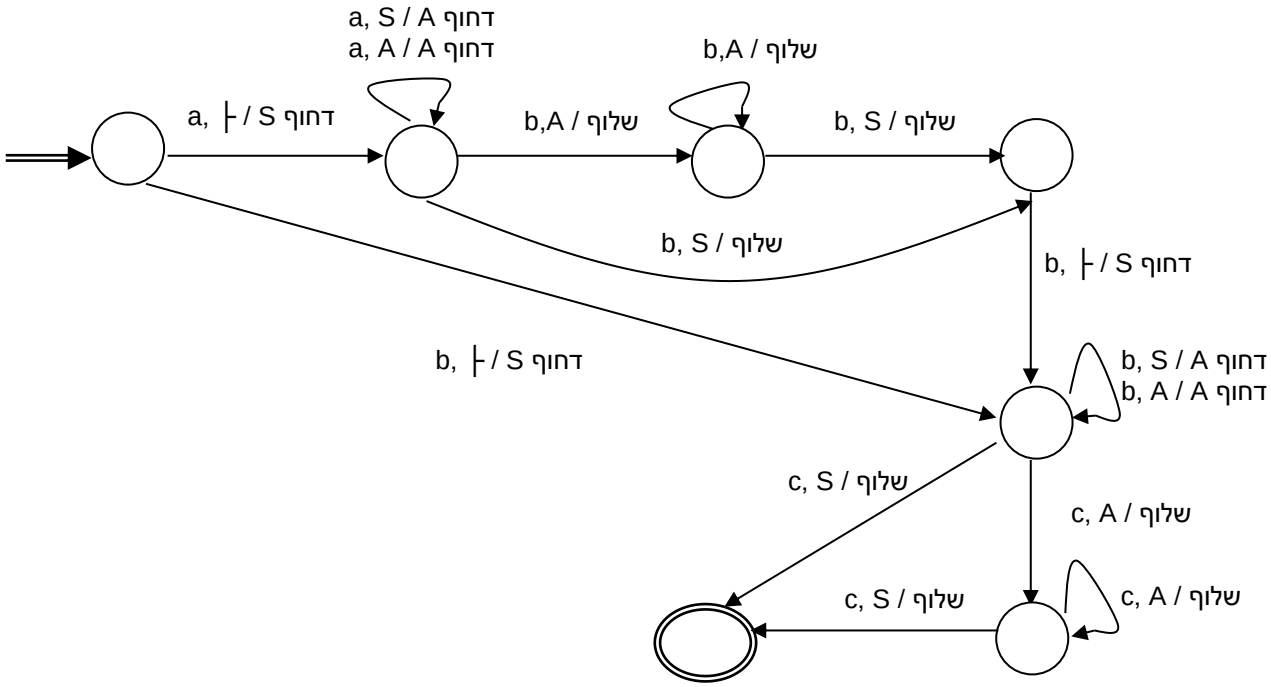
$$w_1 \cdot w = a^i b^j c^{i+j} \in L_1 \ | j \ge 1, i \ge 1$$
$$w_2 \cdot w = a^j b^j c^{i+j} \notin L_1 \mid i, j \ge 1,\ j \ne i \quad \Rightarrow j + j \ne i + j$$

שאלה 16

א. המילה הקצרה היא bc

ב. ניתן להסתכל על השפה L באופן הבא:

$$L = \{a^{i-j}b^{i}c^{j} \mid j \geq 1, i-j \geq 0\} = \{a^{i-j}b^{i-j}b^{j}c^{j} \mid j \geq 1, i-j \geq 0\}$$



פרק ב'

תכנות מונחה עצמים Java

הפתרון לפרק זה נכתב ע"י **אביטל גרינולד EVI**.

תרגיל 17

א. שאלת מעקב, ירושה ופולימורפיזם, זיהוי שגיאות תחביר וריצה

קטע i: תקין, הפלט:

```
A          // בבניה
One of A   // זימון ראשון
One of A   // זימון שני
           //ההמרה אינה משנה דבר
```

קטע ii: שגוי. שגיאת ריצה.

a2 נוצר מהמחלקה A .
בשורה שנייה יש ניסיון להמיר אותו כלפי מטה לטיפוס מהמחלקה B .
שגיאת ריצה. המרות מתבצעות בזמן ריצה.

קטע iii: תקין, הפלט:

```
A          // הפעלת בנאי ריק של A
B          // המשך הבנאי של B
two of B   // פולימורפיזם
one of B   //העצם נוצר כ B
```

קטע iv: שגיאת תחביר

בשורה הראשונה יוצרים עצם מטיפוס A ומציבים אותו במשתנה שהוא הפנייה לעצם מטיפוס B .

קטע v: תקין, הפלט:

```
A          // קודם בניה של A
B          // אח"כ בניה של B
two of B   //העצם נוצר כ B
two of B   //טיפוס העצם קובע איזו פעולה תופעל
```

קטע iv: שגיאת תחביר

בשורה הראשונה יוצרים עצם מטיפוס A ומציבים אותו במשתנה שהוא הפנייה לעצם מטיפוס B .
אין התאמה בטיפוסים. A הוא לא סוג של B.

קטע vii: תקין, הפלט

```
A          // קודם בניה של A
B          // אח"כ בניה של B
one of B   // פולימורפיזם, העצם נוצר כ B
```

ב. השלמת הפעולות הבונות בהתאם לתרשים מעקב

```
// פעולה בונה ריקה
public BBB()
{
        super ();
        this.a = null;
}

// פעולה בונה על פי פרמטר לתכונה
public BBB(AAA a)
{
        super ();
        this.a = a;
}

//A פעולה בונה על פי המספר השלם והטיפוס
public BBB (int n, AAA a)
{
        super (n, n);
        this.a = a;
}

// פעולה בונה על פי כל התכונות
public BBB(int n, double x, AAA a)
{
        super (n, x);
        this.a = a;
}
```

```
// פעולה בונה ריקה
public AAA()
{
        this.n = 0;
        this.x = 0;
}

// פעולה בונה על פי כל התכונות
public AAA (int n, double x)
{
        this.n = n;
        this.x = x;
}
```

אפשר להוסיף בנאי:

```
// פעולה בונה פרמטר אחד
public AAA(int n)
{
        this.n = n;
        this.x = n;
}
```

ואז השורה בבנאי של B תשתנה ל- super (n)

הערות:

(1) חובה להוסיף את הפעולה הבונה הריקה מאחר והוספנו פעולות בונות עם פרמטרים , לכן הפעולה הבונה ברירת המחדל לא קיימת יותר ויש לכתוב אותה באופן מפורש.

(2) בפעולות הבנות אפשר לכתוב מפורשות את ההשמה ואפשר להסתמך על ברירת המחדל שאי הצבת ערך בתכונה שמה ערכי 0 למשתנים פשוטים ו null לעצמים.

(3) בפעולות הבונות של תת המחלקה BBB המשתמשות בבנאי הריק (בנאי ללא פרמטרים) של מחלקת העל AAA, יזומן הבנאי הריק של מחלקת העל AAA גם אם לא נכתוב super()

תרגיל 18

א. ירושה . מעקב ורשום פלט

הפלט הוא:

```
***
C
C
C
CC
---
C
CC
C
D
###
C
C
C
C
CC
C
D
C
D
C
D
C
D
```

כל new C(n) מדפיס n פעמים C.

כל new D(n) מפעיל הבנאים הריקים n פעמים. בכל פעם מודפס C ואחריו D

ב. מעקב כולל תרשים עצמים

count = 0

1) <AA> f1 →

AA
num1 = 10
num2 = 10

count = 1

1) AA ctor1

2) sum1 = 20

3) count = 1

4) <AA> f2 →

AA
num1 = 10
num2 = 20

count = 2

4) AA ctor2

5) count = 2

6) <BB> s1 →

BB
num1 = 1
num2 = 1
num3 = 1

count = 3

6) AA ctor 1

BB ctor 1

7) <AA> f3 →

BB
num1 = 2
num2 = 2
num3 = 2

count = 4

7) AA ctor1

BB ctor1

8) count = 4

9) <AA> f2 →

<BB> s1 →

BB
num1 = 1
num2 = 1
num3 = 1

10) sum = 3 // פולי

12) sum = 4

13) sum = 4 // פולי

הפלט של סעיף ב:

```
AA ctor1
sum1 = 20
count = 1
AA ctor2
count = 2
AA ctor1
BB ctor1
AA ctor1
BB ctor1
count = 4
sum = 3
sum = 4
sum = 4
```

<u>הערה</u>: המשתנה count הוא משתנה מחלקה. יש לו הקצאה אחת בזיכרון. בכל פעם שנוצר עצם חדש, ערכו של count גדל ב 1, כלומר בהתחלה ערכו 0 , אח"כ 1 וכך עד ל 4 .

למען נוחות הכתיבה הוא מופיע 4 פעמים.

תרגיל 19

א. השלמת הטיפוס כך שיתאים להמשך הקוד ויעבור קומפילציה.

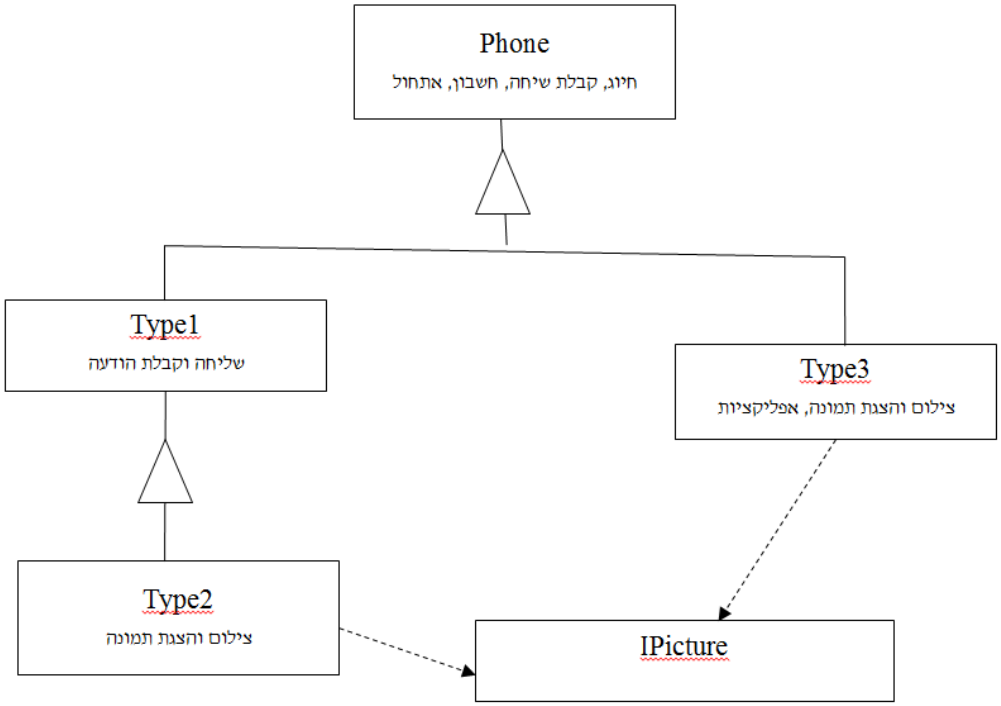
```
Inter1 x = new A();
B y = new B();
Inter3 z = new C();
```

ב. כתיבת פעולה

```
//                                    ט.כ: מערך מטיפוס הממשק Inter1
//        ט.י: הפעלת הפעולה g3 לעצמים עבורם היא קיימת, אחרת תופעל g1
public static void doG (Inter1 [] a)
{
        for (int k=0 ; k < a.length ; k++)
        {
                if (a[k] instanceof A)
                        ((A)a[k]).g3();
                else
                        if (a[k] instanceof C)
                                ( (C)a[k]).g3();
                        else
                                a[k].g1();
        }
}
//---- Inter3 דרך נוספת: g3 מוגדרת בממשק ---
public static void doG (Inter1 [] a)
{
        for (int k=0 ; k < a.length ; k++)
        {
                if (a[k] instanceof Inter3)
                        ((Inter3)a[k]).g3();
                else
                        a[k].g1();
        }
}
```

שאלה 20

א.



ב + ג.

```
public interface IPicture
{
    public void ShowPic (String pic);
    public void takePic (String pic);
}
```

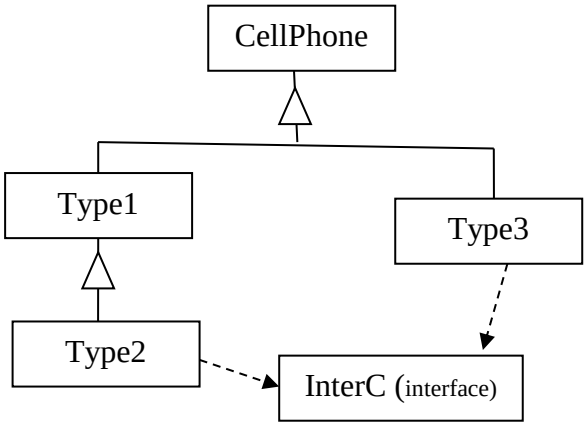
```
public class Phone
{
    protected int minutes;        //  מספר דקות שיחה
    protected String num;         //  מספר המנוי

    public Phone ()
    public int getMinutes()
    public void dial (Phone pn)
    public void recCall (Phone pn)
    public void reset ()

    //--- חשבון עדכני ---
    public double bill()
    {
        return this.minutes * 1.0;
    }
}
```

```
public class Type1 extends Phone
{
    private int mesNum;

    public Type1 ()
    public void reset ()
    public void sendMsg (Phone pn, String st)
    public void recMsg (Phone pn, String st)

    //--- חשבון עדכני ---
    public double bill()
    {
        return super.bill() + this.mesNum * 0.5;
    }
}
public class Type2 extends Type1 implements IPicture
{
    private int picNum;

    public Type2 ()
    public void ShowPic (String pic)
    public void takePic (String pic)
}


public class Type3 extends Phone implements IPicture
{
    private int picNum;
    private int appNum;

    public Type3 ()
    public void ShowPic (String pic)
    public void takePic (String pic)
    public void installApp (String app)
}
```

דרך II:

שימוש ב**מחלקה מופשטת** CellPhone שלא נכתבה בשאלה מאחר ויש תכונות ופעולות שמשותפות לכל טיפוסי מכשירי הטלפון הסלולריים.

המחלקה מופשטת מכיוון שלא יוצרים ממנה עצמים.



ב.

```
public abstract class CellPhone
public static final double PRICE_MINUTE_CONVERSATION = 1; // תעריף דקת שיחה
protected int minutes;  // מספר דקות שיחה
protected String phoneNumber;   // מספר טלפון
public CellPhone (String phoneNumber)        // פעולה בונה טלפון על פי מספרו
public void dial (String phoneNumber, int minutes)  // חיוג למספר טלפון עם מספר דקות השיחה
public void recCall(String phoneNumber)      // קבלת שיחה ממספר טלפון
public void reset()      // איפוס דקות השיחה
```

```
public class Type1 extends CellPhone
protected int numOfMessages; // מספר הודעות
public Type1(String phoneNumber) // בניית טלפון מסוג 1 על פי מספר טלפון
public void senMsg(String phoneNumber) //שליחת מסרון למספר טלפון
public void recMsg(String phoneNumber) // קבלת מסרון ממספר טלפון
public void reset() // דריסת הפעולה ממחלקת העל:איפוס דקות שיחה ומספר ההודעות
```

```
public class Type2 extends Type1 implements InterC

public Type2 (String phoneNumber)          // בניית טלפון מסוג 2 על פי מספר טלפון
public void takePic(String pic)          // צילום תמונה
public void showPic(String pic)          // הצגת תמונה
```

```
public class Type3 extends CellPhone implements InterC

public Type3 (String phoneNumber)        // בניית טלפון מסוג 3 על פי מספר טלפון
public void takePic (String pic)         // צילום תמונה
public void showPic (String pic)         // הצגת תמונה
```

ג.

במחלקה **CellPhone**

```
// החזרה חשבון טלפון עדכני
public double bill ()
{
        return  this.minutes * CellPhone.PRICE_MINUTE_CONVERSATION;
}
```

במחלקה **Type1**

```
// החזרה חשבון טלפון עדכני
public double bill()
{
        return super.bill() + this.numOfMessages * Type1.PRICE_SMS;
}
```

הסבר למבנה היררכית המחלקות והממשקים:

1- אין אף טלפון שיש לו תכונות ופעולות המשותפות לשלושת טיפוסי הטלפון, לכן, חייבים ליצור מחלקת על שתכלול את הפעולות והתכונות המשותפות לכולם. מכיוון שזו מחלקה שלא יוצרים ממנה עצמים, היא צריכה להיות מופשטת.
מי שלא למד מחלקות מופשטות, יכול לעשות מחלקה "רגילה".

2- התכונות המשותפות לכולם: מספר דקות שיחה.

3- הפעולות המשותפות לכולם: חיוג, קבלת שיחה, איפוס ועריכת חשבון לתשלום.

4- מספר ההודעות והפעולות בטפול ההודעות משותפות למכשירים מסוג 1+2

מחלקה CellPhon

5- למכשיר 2 יש פעולות נוספות על אלו שבמכשיר 2 ולכן הוא הרחבה של הטיפוס.

6- למכשיר 2 ו 3 פעולות משותפות המטפלות בתמונה אבל אי אפשר ליצור היררכית ירושה ביניהם ולכן המשותף בינהם הוא התפקוד.

7- למכשיר 3 טיפול נוסף ייחודי עבורו, טיפול ביישומים.

ראה הטבלה הבאה:

Group	Type1	טלפון 1
A	dial	חיוג
	recCall	קבלת שיחה
B	sendMsg	שליחת הודעה
	recNsg	קבלת הודעה
	Type2	**טלפון 2**
A	dial	חיוג
	recCall	קבלת שיחה
B	sendMsg	שליחת הודעה
	recMsg	קבלת הודעה
C	takePic	צילום תמונה
	showPic	הצגת תמונה
	Type3	**טלפון 3**
A	dial	חיוג
	recCall	קבלת שיחה
C	takePic	צילום תמונה
	showPic	הצגת תמונה
D	installApp	התקנת אפליקציות
	Common	**משותפות**
Com	reset	איפוס
	bill	חשבון
	minutes	דקות ש'(תכונה)

תכנות מונחה עצמים C#

הפתרון לפרק זה נכתב ע"י **אביטל גרינולד EVI ועקיבא לביא**

תרגיל 21

א. שאלת מעקב, ירושה ופולימורפיזם, זיהוי שגיאות תחביר וריצה

קטע i: תקין, הפלט:

A	// בבניה
One of A	// זימון ראשון
One of A	// זימון שני
	//המרה אינה משנה דבר

קטע ii: שגוי. שגיאת ריצה.

a2 נוצר מהמחלקה A .
בשורה שנייה יש ניסיון להמיר אותו כלפי מטה לטיפוס מהמחלקה B .
שגיאת ריצה. המרות מתבצעות בזמן ריצה.

קטע iii: תקין, הפלט:

A	// הפעלת בנאי ריק של A
B	// המשך הבנאי של B
two of B	// פולימורפיזם
one of B	//העצם נוצר כ B

קטע iv: שגיאת תחביר

בשורה הראשונה יוצרים עצם מטיפוס A ומציבים אותו במשתנה שהוא הפנייה לעצם מטיפוס B .

קטע v: תקין, הפלט:

A	// קודם בניה של A
B	// אח"כ בניה של B

קטע iv: שגיאת תחביר

בשורה הראשונה יוצרים עצם מטיפוס A ומציבים אותו במשתנה שהוא הפנייה לעצם מטיפוס B .
אין התאמה בטיפוסים. A הוא לא סוג של B.

קטע vii: תקין, הפלט

A	// קודם בניה של A
B	// אח"כ בניה של B
one of B	// פולימורפיזם, העצם נוצר כ B

ב. השלמת הפעולות הבונות בהתאם לתרשים מעקב

```
// פעולה בונה ריקה
public BBB()
{
        super ();
        this.a = null;
}

// פעולה בונה על פי פרמטר לתכונה
public BBB(AAA a)
{
        super ();
        this.a = a;
}


//A פעולה בונה על פי המספר השלם והטיפוס
public BBB (int n, AAA a)
{
        super (n, n);
        this.a = a;
}

// פעולה בונה על פי כל התכונות
public BBB(int n, double x, AAA a)
{
        super (n, x);
        this.a = a;
}
```

```
// פעולה בונה ריקה
public AAA()
{
        this.n = 0;
        this.x = 0;
}

// פעולה בונה על פי כל התכונות
public AAA (int n, double x)
{
        this.n = n;
        this.x = x;
}
```

<u>הערות:</u>

(1) חובה להוסיף את הפעולה הבונה הריקה מאחר והוספנו פעולות בונות עם פרמטרים , לכן הפעולה הבונה ברירת המחדל לא קיימת יותר ויש לכתוב אותה באופן מפורש.

(2) בפעולות הבנות אפשר לכתוב מפורשות את ההשמה ואפשר להסתמך על ברירת המחדל שאי הצבת ערך בתכונה שמה ערכי 0 למשתנים פשוטים ו null לעצמים.

(3) בפעולות הבונות במחלקה הבת BBB , אפשר לכתוב ()super כאשר מזמנים את את בנאי מחלקת העל הריק או לא. בכל מקרה יזמן אותו בראש הפעולה הבונה.

תרגיל 22

א. ירושה . מעקב ורשום פלט

הפלט הוא:

```
***
C
C
C
CC
---
C
CC
C
D
###
C
C
C
C
CC
C
D
C
D
C
D
C
D
```

כל new C(n) מדפיס n פעמים C.

כל new D(n) מפעיל הבנאים הריקים n פעמים. בכל פעם מודפס C ואחריו D

ב. מעקב כולל תרשים עצמים

count = 0

2) <AA> f1 →

AA
num1 = 10
num2 = 10

count = 1

4) AA ctor1

5) sum1 = 20

6) count = 1

4) <AA> f2 →

AA
num1 = 10
num2 = 20

count = 2

4) AA ctor2

5) count = 2

6) <BB> s1 →

BB
num1 = 1
num2 = 1
num3 = 1

count = 3

6) AA ctor 1

BB ctor 1

7) <AA> f3 →

BB
num1 = 2
num2 = 2
num3 = 2

count = 4

7) AA ctor1

BB ctor1

8) count = 4

9) <AA> f2 →

<BB> s1 →

BB
num1 = 1
num2 = 1
num3 = 1

10) sum = 3 // פולי

12) sum = 4

13) sum = 4 // פולי

הפלט של סעיף ב:

```
AA ctor1
sum1 = 20
count = 1
AA ctor2
count = 2
AA ctor1
BB ctor1
AA ctor1
BB ctor1
count = 4
sum = 3
sum = 4
sum = 4
```

<u>הערה</u>: המשתנה count הוא משתנה מחלקה. יש לו הקצאה אחת בזיכרון. בכל פעם שנוצר עצם חדש, ערכו של count גדל ב 1, כלומר בהתחלה ערכו 0 , אח"כ 1 וכך עד ל 4 .

למען נוחות הכתיבה הוא מופיע 4 פעמים.

תרגיל 23

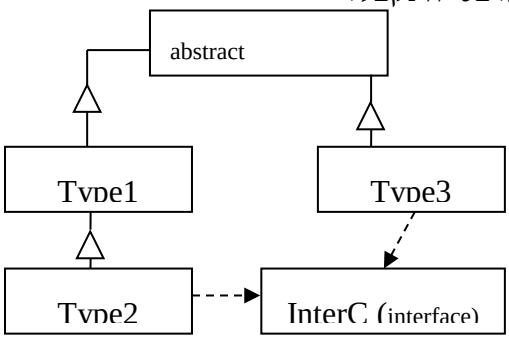
שאלה 24

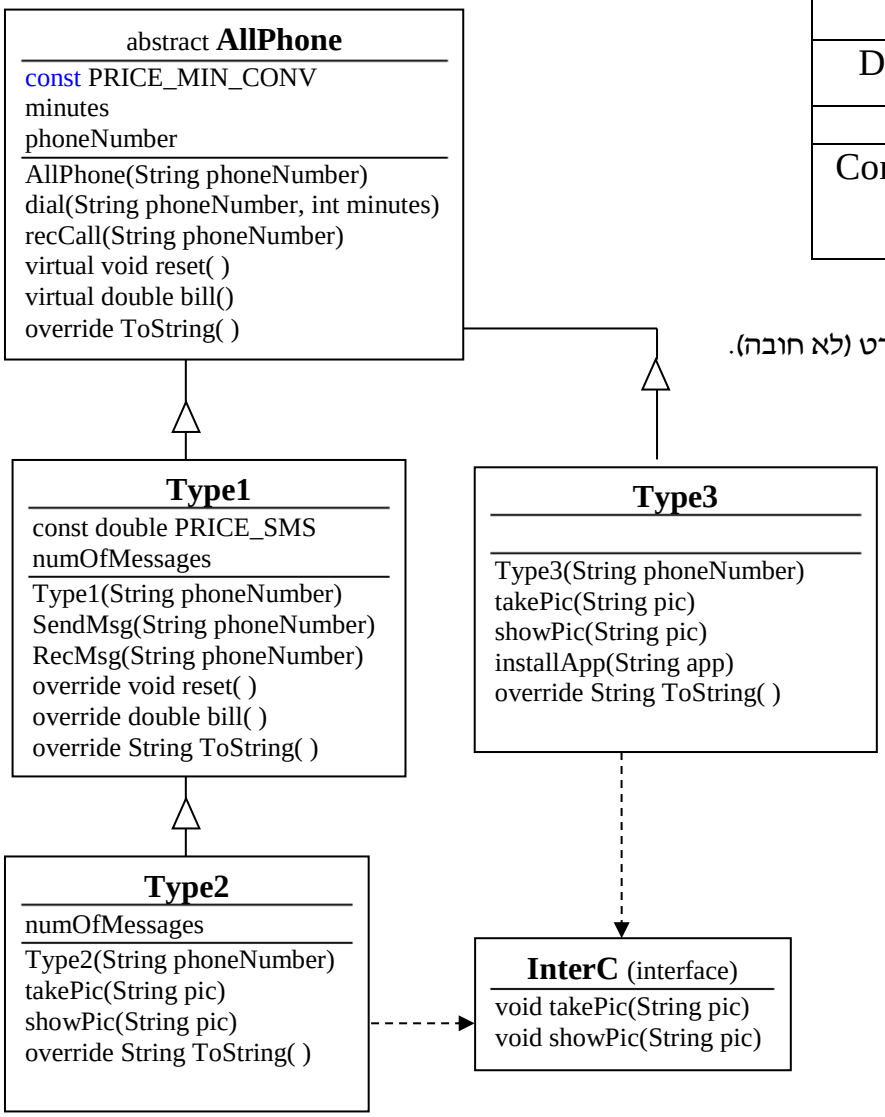
א. ניתוח: נחלק את הפעולות לתת-קבוצות לפי הפעולות.

Group	Type1	טלפון 1
A	Dial	חיוג
	RecCall	קבלת שיחה
B	SendMsg	שליחת הודעה
	RecMsg	קבלת הודעה
	Type2	**טלפון 2**
A	Dial	חיוג
	RecCall	קבלת שיחה
B	SendMsg	שליחת הודעה
	RecMsg	קבלת הודעה
C	TakePic	צילום תמונה
	ShowPic	הצגת תמונה
	Type3	**טלפון 3**
A	Dial	חיוג
	RecCall	קבלת שיחה
C	TakePic	צילום תמונה
	ShowPic	הצגת תמונה
D	InstallApp	התקנת אפליקציות
	Common	**משותפות**
Com	Reset	איפוס
	Bill	חשבון
	min	דקות ש'(תכונה)

הרכב הקבוצות הוא כזה, שאין אף טלפון שיכול להיות מחלקה עליונה מכיוון לכל אחד מהשלושה פעולות שאינן בטלפון אחר. לכן, חייבים ליצור מחלקת על שתכלול את הפעולות והתכונות המשותפות לכולם. מכיוון שזו מחלקה שלא יוצרים ממנה עצמים, היא צריכה להיות מופשטת. מי שלא למד מחלקות מופשטות, יכול לעשות מחלקה "רגילה".

קבוצות A, Com ו- min משותפת לשלשת המחלקות, הן תהיינה במחלקה העליונה. בס"ה נקבל:





<u>סעיף ב</u>: יהיה יותר קל עם UML מפורט (לא חובה).
כך רואים יותר טוב היכן ואיך לממש.
פרוט ב #C בעמוד הבא.

פרוט בקוד #C

```
54  using System;
55  namespace Q20_24
56  {
57     public interface InterC
58     {
59        void takePic(String pic);                    // נניח שהמחרוזת היא שם התמונה
60        void showPic(String pic);
61     }
62
63     public abstract class AllPhone
64     {
65        protected const double PRICE_MIN_CONV = 1;
66        protected int minutes;                                  // דקות שיחה
67        protected String phoneNumber;                           // מספר הטלפון
68        public AllPhone(String phoneNumber)  // פעולה בונה. הפרמטר: מספר הטלפון
69        {
70           this.minutes = 0;
71           this.phoneNumber = phoneNumber;
72        }
73        /* dial to phoneMuner and talk minutes minutes
74         * @param phoneNumber: String
75         * @param minutes : int     */
76        public void dial(String phoneNumber, int minutes)
77        { this.minutes += minutes; }
78        /* get conversation from phoneNumber
79         * @param phoneNumber : String     */
80        public void recCall(String phoneNumber)
81        { }
82        /* zeroing the minutes     */
83        public virtual void reset()
84        { this.minutes = 0; }
85        /* calculate bill
86         * @return bill     */
87        public virtual double bill( )                // סעיף ג: פעולות חישוב החשבון
88        { return this.minutes * PRICE_MIN_CONV; }
89        public override String ToString()
          { return "CellPhone: phneNumber is: " + this.phoneNumber +
                   " minutes = " + this.minutes;   }
       }
```

```
public class Type1 : AllPhone
{
   public const double PRICE_SMS = 0.5;
   protected int numOfMessages;
   public Type1(String phoneNumber) : base(phoneNumber)
   { this.numOfMessages = 0; }
   public void senMsg(String phoneNumber)
   { this.numOfMessages++; }
   public void recMsg(String phoneNumber)
   { }
   public override void reset( )
   {
      base.reset( );
      this.numOfMessages = 0;
   }
   public override double bill( )                    // סעיף ג: פעולות חישוב החשבון
   { return base.bill( ) + this.numOfMessages * PRICE_SMS; }
   public override String ToString( )
   {
      return "Type1 is kind of " + base.ToString() + "number of sms = " +
         this.numOfMessages + " act as InterB";
   }
}

public class Type2 : Type1, InterC
{
   public Type2(String phoneNumber)  : base(phoneNumber)
   {  }
   public void takePic(String pic)
   { }
   public void showPic(String pic)
   { }
   public override String ToString( )
   {
      return "Type2 is kind of " + base.ToString( ) +
      " with pictures act as  InterC";
   }
}
```

```
public class Type3 : AllPhone, InterC
  {
    public Type3(String phoneNumber) : base(phoneNumber)
    { }
    public void takePic(String pic)
    { }
    public void showPic(String pic)
    { }
    public void installApp(String app)
    { }
    public override String ToString()
    { return "Type3 is kind of " + base.ToString() + " act as InterC"; }
  }

  class Program
  {
    static void Main(string[] args) // === התכנית הראשית
    {
      AllPhone[] ar = new AllPhone[3];
      ar[0] = new Type1("aa");
      ar[0].dial("cc", 20);
      ((Type1)ar[0]).senMsg("bb");
      ((Type1)ar[0]).senMsg("cc");
      ((Type1)ar[0]).senMsg("dd");
      ar[1] = new Type2("bb");
      ar[1].dial("dd", 25);
      ((Type2)ar[1]).takePic("pic1");
      ((Type1)ar[1]).senMsg("aa");
      ar[2] = new Type3("cc");
      ar[2].dial("ada", 25);
      for (int k = 0; k < ar.Length; k++)
      {
        Console.WriteLine(ar[k]);
        Console.WriteLine("bill is: " + ar[k].bill());
      }
      Console.WriteLine("\nEnd"); Console.ReadLine();
    }
  }
}
```