

מדעי המחשב ב'

פתרון בחינת הקטרת

פרק א - עיצוב תכנה

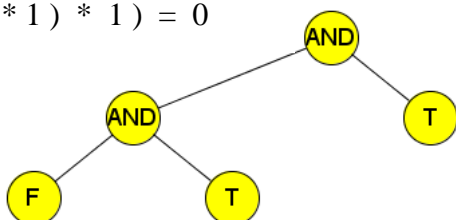
שאלה 1

i - false

.א

$(F \text{ and } T) \text{ and } T = F$

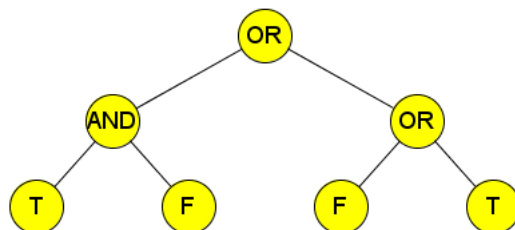
$(0 * 1) * 1 = 0$



ii - true

$(T \text{ and } F) \text{ OR } (F \text{ OR } T) = T$

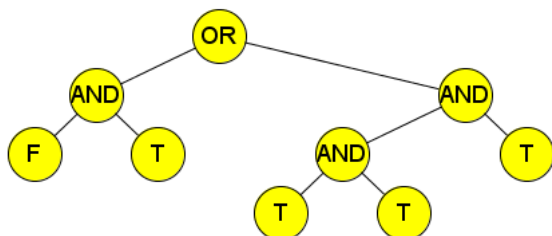
$(1 * 0) + (0 + 1) =$
 $0 + 1 = 1$



iii - true

$(T \text{ and } F) \text{ OR } ((T \text{ AND } T) \text{ AND } T) = T$

$(1 * 0) + ((1 * 1) * 1) =$
 $0 + 1 * 1 = 0 + 1 = 1$



True	1
False	0
AND	*
OR	+

עץ-ביטוי-בוליאני (t)

אם עלה? (t)

אם ערך העלה שווה T החזר אמת

אחרת - החזר שקר

אחרת -

אם ערך הצומת "וגם"

החזר עץ-ביטוי-בוליאני (בן שמאלי של t) וגם עץ-ביטוי-בוליאני (בן ימני של t)

אחרת - החזר עץ-ביטוי-בוליאני (בן שמאלי של t) או עץ-ביטוי-בוליאני (בן ימני של t)

Java:

```
//--- פעולה המחזירה אמת אם הצומת הוא עלה ושקר אחרת ---
public static boolean isLeaf(BinTreeNode<String> t)
{
    if (t == null) return false;
    if (t.getLeft() == null && t.getRight() == null)
        return true;
    return false;
}

//--- פעולה המחזירה את עץ הביטוי הבוליאני ---
public static boolean boolTreeExp (BinTreeNode<String> bt)
{
    if (isLeaf(bt))
    {
        if (bt.getInfo().equals("T"))
            return true;
        return false;
    }
    if (bt.getInfo().equals("AND"))
        return boolTreeExp(bt.getLeft()) && boolTreeExp(bt.getRight());
    return boolTreeExp(bt.getLeft()) || boolTreeExp(bt.getRight());
}
```

C#:

```
//--- פעולה המחזירה אמת אם הצומת הוא עלה ושקר אחרת ---
public static bool IsLeaf(BinTreeNode<string> t)
{
    if (t == null) return false;
    if (t.GetLeft() == null && t.GetRight() == null)
        return true;
    return false;
}

//--- פעולה המחזירה את עץ הביטוי הבוליאני ---
public static bool BoolTreeExp(BinTreeNode<string> bt)
{
    if (IsLeaf(bt))
    {
        if (bt.GetInfo() == "T") // if (bt.GetInfo().Equals("T"))
            return true;
        return false;
    }
    if (bt.GetInfo() == "AND")
        return BoolTreeExp(bt.GetLeft()) && BoolTreeExp(bt.GetRight());
    return BoolTreeExp(bt.GetLeft()) || BoolTreeExp(bt.GetRight());
}
```

פתרון Java לפי התכנית החדשה:

```
//--- פעולה המחזירה אמת אם הצומת הוא עלה ושקר אחרת ---
public static boolean isLeaf(BinNode<String> t)
{
    if (t == null) return false;
    if (t.getLeft() == null && t.getRight() == null)
        return true;
    return false;
}

//--- פעולה המחזירה את ערך עץ הביטוי הבוליאני ---
public static boolean boolTreeExp (BinNode<String> bt)
{
    if (isLeaf(bt))
    {
        if (bt.getValue().equals("T"))
            return true;
        return false;
    }
    if (bt.getValue().equals("AND"))
        return boolTreeExp(bt.getLeft()) && boolTreeExp(bt.getRight());
    return boolTreeExp(bt.getLeft()) || boolTreeExp(bt.getRight());
}
```

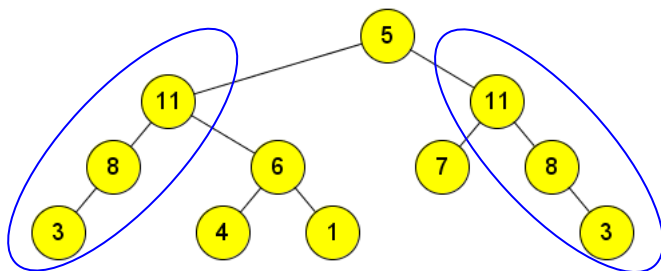
פתרון C# לפי התכנית החדשה - נכתב ע"י ראמי ג'באלי:

```
//פעולה מחזירה אמת אצומת הוא עלה אחרת הפעלה מחזירה שקר/
public static bool IsLeaf(BinNode<string> t)
{
    return (t.GetLeft() == null && t.GetRight() == null);
}

//פעולה מחזירה הערך הבוליאני של הביטוי/
public static bool Check(BinNode<string> t)
{
    if (IsLeaf(t) && t.GetValue() == "T")
        return true;
    if (IsLeaf(t) && t.GetValue() == "F")
        return false;
    if (t.GetValue() == "AND")
        return Check(t.GetLeft()) && Check(t.GetRight());
    return Check(t.GetLeft()) || Check(t.GetRight());
}
```

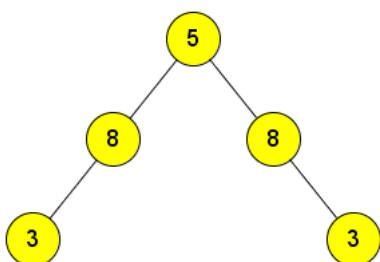
שאלה 2

א. (1) עבור העץ הנתון מחזירה הפעולה **אמת**

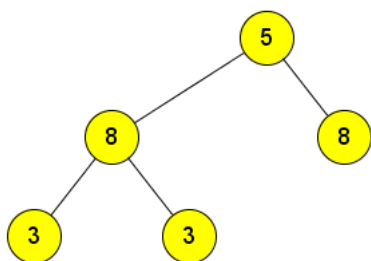


הפעולה מחזירה **אמת** אם כל צומת בענף השמאלי של העץ שווה בערכו לצומת באותה רמה בענף הימני של העץ, ושקר אחרת.

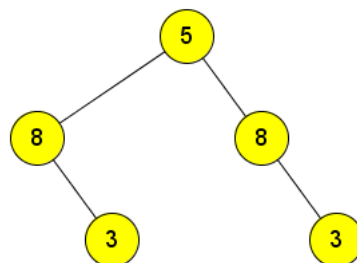
(2) עבור העץ שלהלן תחזיר הפעולה **אמת**



עבור העצים שלהלן תחזיר הפעולה **שקר**



תנאי 2: חסר בן ימני ל-8 בענף הימני



תנאי 2: חסר בן שמאלי ל-8 בענף השמאלי

ב. (1)

	0	1	3	3
a	5	7	8	12

n	i	i < 3	a[i]	a[i+1]	a[i] > a[i+1]
4	0	כן	5	7	לא
	1	כן	7	8	לא
	2	כן	8	12	לא
	3	לא			

(2) סיבוכיות הפעולה $O(n)$ כי נוגעים בכל אחד מ-n תאי המערך בדיוק פעם אחת

(3)

שלוש השורות שלהלן מבצעות החלפה
בין ערכי התאים שבמקומות i ו- i+1 במערך

$a[i] = a[i] + a[i+1]$
 $a[i+1] = a[i] - a[i+1]$
 $a[i] = a[i] - a[i+1]$

n	i	i < 3	a[i]	a[i+1]	a[i] > a[i+1]	תאים מוחלפים	a	0	1	2	3
4	0	כן	12	8	כן	a[0], a[1]		12	8	7	5
	0	כן	8	12	לא			8	12	7	5
	1	כן	12	7	כן	a[1], a[2]		8	12	7	5
	0	כן	8	7	כן	a[0], a[1]		8	7	12	5
	0	כן	7	8	לא			7	8	12	5
	1	כן	8	12	לא			7	8	12	5
	2	כן	12	5	כן	a[2], a[3]		7	8	12	5
	0	כן	7	8	לא			7	8	5	12
	1	כן	8	5	כן	a[1], a[2]		7	8	5	12
	0	כן	7	5	כן	a[0], a[1]		7	5	8	12
	0	כן	5	7	לא			5	7	8	12
	1	כן	7	8	לא						
	2	כן	8	12	לא			המערך בסיום:			
	3	לא						5	7	8	12

החלפת האיבר במקום ה- k גוררת מיון מחודש של כל האיברים שלפניו ביעילות ריבועית (כמו מיון בועות):
 $1^2 + 2^2 + 3^2 + \dots + n^2 \Rightarrow \frac{1}{2}(1^2 + n^2) * n \Rightarrow O(n^3)$

סיבוכיות הפעולה $O(n^3)$.
 עבור כל אחד מ- n האיברים הלא ממויינים,
 מבצעים מיון בסד"ג $O(n^2)$.
 $f(n) = n * n^2 \Rightarrow O(n^3)$.

: Java

שאלה 3

```
public class BusRoute
{
    private List<Station>path;          // רשימת התחנות במסלול

    public BusRoute (Station first, Station second)
    {
        this.path = new List<Station>();
        this.path.insert(null, second);
        this.path.insert(null, first);
    }

    //--- הוספת תחנה לסוף המסלול ---
    public void addStation (Station newStation)
    {
        Node <Station> p = null;
        Node <Station> pos = this.path.getFirst();
        while (pos != null)
        {
            p = pos;
            pos = pos.getNext();
        }
        this.path.insert(p, newStation);
    }

    //--- חישוב אורך המסלול ---
    public double routeLength()
    {
        double len = 0;
        Node <Station> pos = this.path.getFirst();
        if (pos == null)
            return 0;
        while (pos.getNext() != null)
        {
            len += pos.getInfo().distance(pos.getNext().getInfo());
            pos = pos.getNext();
        }
        return len;
    }

    public String toString()
    {
        String str = this.path.toString();
        str = str.replace(" , ", " --> ");
        return str;
    }
}
```

```
public class Q3_BusRoute
{
    public static void main(String[] args)
    {
        Station first = new Station (0,2);
        Station second = new Station (1,4);

        BusRoute br = new BusRoute (first, second);
        System.out.println(br);
        System.out.println(br.routeLength());
        System.out.println();

        br.addStation(new Station (5,4));
        br.addStation(new Station (3,1));
        br.addStation(new Station (5,0));

        System.out.println(br);
        System.out.println(br.routeLength());
    }
}
```

```
/*
```

```
[(0,2) --> (1,4)]
2.23606797749979
```

```
[(0,2) --> (1,4) --> (5,4) --> (3,1) --> (5,0)]
12.07768723046357
```

```
*/
```

פתרון Java לפי התכנית החדשה:

```
public class BusRoute
{
    //--- פתרון לפי תכנית הלימודים החדשה ---
    //--- רשימת התחנות במסלול מיוצגת בשרשרת חוליות ---
    private Node<Station>path;

    public BusRoute (Station first, Station second)
    {
        this.path = new Node<Station>(second);
        this.path = new Node<Station> (first, this.path);
    }

    //--- הוספת תחנה לסוף המסלול ---
    public void addStation (Station newStation)
    {
        Node <Station> p = null;
        Node <Station> pos = this.path;
        while (pos != null)
        {
            p = pos;
            pos = pos.getNext();
        }
        p.setNext(new Node<Station>(newStation));
    }

    //--- חישוב אורך המסלול ---
    public double routeLength()
    {
        double len = 0;
        Node <Station> pos = this.path;
        if (pos == null)
            return 0;
        while (pos.getNext() != null)
        {
            len += pos.getInfo().distance(pos.getNext().getInfo());
            pos = pos.getNext();
        }
        return len;
    }

    public String toString()
    {
        String str = this.path.toString();
        Node<Station>pos = this.path.getNext();
        while (pos != null)
        {
            str += " --> " + pos.getInfo().toString();
            pos = pos.getNext();
        }
        return str;
    }
}
```


C#:

עלה 3

```
class BusRoute
{
    private List<Station> path;    // רשימת התחנות במסלול

    public BusRoute(Station first, Station second)
    {
        this.path = new List<Station>();
        this.path.Insert(null, second);
        this.path.Insert(null, first);
    }

    ///--- הוספת תחנה לסוף המסלול ---
    public void AddStation(Station newStation)
    {
        Node<Station> p = null;
        Node<Station> pos = this.path.GetFirst();
        while (pos != null)
        {
            p = pos;
            pos = pos.GetNext();
        }
        this.path.Insert(p, newStation);
    }

    ///--- חישוב אורך המסלול ---
    public double RouteLength()
    {
        double len = 0;
        Node<Station> pos = this.path.GetFirst();
        if (pos == null)
            return 0;
        while (pos.GetNext() != null)
        {
            len += pos.GetInfo().Distance(pos.GetNext().GetInfo());
            pos = pos.GetNext();
        }
        return len;
    }

    public override string ToString()
    {
        string str = this.path.ToString();
        str = str.Replace(" , ", " --> ");
        return str;
    }
}
```

```
class Q3_BusRoute
{
    static void Main(string[] args)
    {
        Station first = new Station (0,2);
        Station second = new Station (1,4);

        BusRoute br = new BusRoute (first, second);
        Console.WriteLine(br);
        Console.WriteLine(br.RouteLength());
        Console.WriteLine(); /*
                                [(0,2) --> (1,4)]
                                2.23606797749979
                                [(0,2) --> (1,4) --> (5,4) --> (3,1) --> (5,0)]
                                12.0776872304636
                                */

        br.AddStation(new Station (5,4));
        br.AddStation(new Station (3,1));
        br.AddStation(new Station (5,0));

        Console.WriteLine(br);
        Console.WriteLine(br.RouteLength());
    }
}
```

פתרון C# לפי התכנית החדשה - נכתב ע"י ראמי ג'באלי:

```
//--- המחלקה מסלול אוטובוס לפי תכנית הלימודים החדשה ---
//--- מסלול האוטובוס מיוצג ע"י שרשרת חוליות ---
//--- נכתב ע"י ראמי ג'באלי ---
public class BusRoute
{
    private Node<Station> path;
    public BusRoute(Station first, Station second)
    {
        Node<Station> p = new Node<Station>(second);
        this.path = new Node<Station>(first, p);
    }
    //פעולה מוסיפה תחנה לסוף
    public void AddStaion(Station newSttion)
    {
        Node<Station> pos = this.path;
        while (pos.GetNext() != null)
        {
            pos = pos.GetNext();
        }
        Node<Station> temp = new Node<Station>(newSttion, pos.GetNext());
        pos.SetNext(temp);
    }
}
```

```
//פעולה מחזירה אורך מסלול
public double RouteLength()
{
    double sum = 0;
    Node<Station> pos = this.path;
    while (pos.GetNext() != null)
    {
        sum = sum + pos.GetValue().Distance(pos.GetNext().GetValue());
        pos = pos.GetNext();
    }
    return sum;
}

//פעולה מדפיסה השרשת
public void Print()
{
    Node<Station> p = this.path;
    while (p != null)
    {
        Console.Write(p.GetValue()+"-->");
        p = p.GetNext();
    }
    Console.WriteLine("null");
}

}

static void Main(string[] args)
{
    Station first = new Station(0, 2);
    Station second = new Station(1, 4);
    BusRoute busroute = new BusRoute(first, second);

    busroute.AddStaion(new Station(5, 4));
    busroute.AddStaion(new Station(3, 1));
    busroute.AddStaion(new Station(5, 0));

    busroute.Print();

    Console.WriteLine(busroute.RouteLength());
}
```

:Java

עלף 4

```

/*
 *      א. המחלקה סדרן משימות
 */
public class TaskManagment
{
    private Stack<Task> s0;      // מחסנית לשמירת משימות בהולות
    private Stack<Task> s1;      // מחסנית לשמירת משימות דחופות
    private Queue<Task> q2;      // תור לשמירת משימות רגילות

    //--- ב. פעולה בונה ---
    public TaskManagment ()
    {
        this.s0 = new Stack<Task>();
        this.s1 = new Stack<Task>();
        this.q2 = new Queue<Task>();
    }

    //--- ג. הוספת משימה לסדרן המשימות ---
    public void addTask (Task ts)
    {
        int code = ts.getCode();
        switch (code)
        {
            case 0: this.s0.push(ts); break;
            case 1: this.s1.push(ts); break;
            case 2: this.q2.insert(ts);
        }
    }

    //--- ד. הוצאת משימה לביצוע לפי מידת הדחיפות ---
    public Task removeTask ()
    {
        if (! this.s0.isEmpty())
            return s0.pop();
        if (! this.s1.isEmpty())
            return s1.pop();
        if (! this.q2.isEmpty())
            return q2.remove();
        return null;
    }
}

```

C#:

עמוד 4

```
class TaskManagment
{
    private Stack<Task> s0;    // מחסנית לשמירת משימות בהולות
    private Stack<Task> s1;    // מחסנית לשמירת משימות דחופות
    private Queue<Task> q2;    // תור לשמירת משימות רגילות

    //--- ב. פעולה בונה ---
    public TaskManagment()
    {
        this.s0 = new Stack<Task>();
        this.s1 = new Stack<Task>();
        this.q2 = new Queue<Task>();
    }

    //--- ג. הוספת משימה לסדרן המשימות ---
    public void AddTask(Task ts)
    {
        int code = ts.GetCode();
        switch (code)
        {
            case 0: this.s0.Push(ts); break;
            case 1: this.s1.Push(ts); break;
            case 2: this.q2.Insert(ts); break;
        }
    }

    //--- ד. הוצאת משימה לביצוע לפי מידת הדחיפות ---
    public Task RemoveTask()
    {
        if (!this.s0.IsEmpty())
            return s0.Pop();
        if (!this.s1.IsEmpty())
            return s1.Pop();
        if (!this.q2.IsEmpty())
            return q2.Remove();
        return null;
    }
}
```

פרק ב'

מערכות מחשב ואסמבלר
הפתרון לפרק זה נכתב ע"י: **רונית מרציאנו**

תרגיל 5

א.

BX		CX		AX	
BH	BL	CH	CL	AH	AL
30	00h		05h		05h
30	01h		04h		04h
			03h		03h
			02h		02h
			01h		01h
					00h
03h				תוכן הזיכרון	
3000h		3001h		כתובת	

ב.

BX		CX		AX	
BH	BL	CH	CL	AH	AL
30	00h		05h		05h
30	01h		04h		04h
30	02h		03h		03h
			02h		02h
			01h		01h
					00h
05h		03h		תוכן הזיכרון	
3000h		3001h		3002h	
				כתובת	

ג.

BX		CX		AX	
BH	BL	CH	CL	AH	AL
30	00h		05h		05h
30	01h		04h		04h
			03h		03h
			02h		02h
			01h		01h
					00h
03h				תוכן הזיכרון	
3000h		3001h			
				כתובת	

```

MOV BX,3000H
MOV CX,5
AGAIN:
MOV AL,CL
CMP AL,3
JNZ NEXT
MOV [BX],CL
INC BX
NEXT:
LOOPZ AGAIN
SHOOV:
MOV AL,CL
CMP AL,3
JNZ NEXT1
MOV [BX],CL
INC BX
NEXT1:
LOOP SHOOV
NOP
    
```

```

MOV BX,3000H
MOV CX,5
AGAIN:
MOV AL,CL
CMP AL,3

MOV [BX],CL
INC BX
NEXT:
LOOPZ AGAIN
SHOOV:
MOV AL,CL
CMP AL,3
JNZ NEXT1
MOV [BX],CL
INC BX
NEXT1:
LOOP SHOOV
NOP
    
```

```

MOV BX,3000H
MOV CX,5
AGAIN:
MOV AL,CL
CMP AL,3
JNZ NEXT
MOV [BX],CL
INC BX
NEXT:
LOOPZ AGAIN
SHOOV:
MOV AL,CL
CMP AL,3
JNZ NEXT1
MOV [BX],CL
INC BX
NEXT1:
LOOPNE SHOOV
NOP
    
```

תרגיל 6

א.

הוראות	CF	ZF	SF
MOV AL,3H	-	-	-
CMP AL,3H	-	+	-
CMP AL,2H	-	-	-
CMP AL,5H	+	-	+
XOR AL,AL	-	+	-
DEC AL	-	-	+
MUL BL	-	-	+

ב. (1) (2)

```
XOR SI,SI ;index in ARR
XOR DX,DX ; counter
MOV CX,5 ; lenght of ARR
XOR AX,AX
```

SHOOV :

```
PUSH CX ; using loop twice
MOV AL,ARR[SI] ; next number from ARR
PUSH AX ;ax push to stack - parameter
CALL PEULA
POP AX ; get answer from stack
TEST AX,1 ; check answer
JZ CONT
INC DX
```

CONT:

```
INC SI
POP CX
LOOP SHOOV
```

```
MOV CX,DX ; answer in cx
```

PEULA PROC

```
MOV BP,SP
MOV AX,[BP+2]
MOV CX,8 ; number is between 0-255 8bits sufficient
XOR BX,BX ; counter
```

AGAIN:

```
SHR AL,1
ADC BL,0
LOOP AGAIN
MOV [BP+2],BX
RET
```

ENDP

תרגיל 7

א.

- (1) `PUSH AL` → שגוי
`POP AH` הסבר: אפשר לדחוף למחסנית רק מילים
- (2) תקין
 בשיטת המשלים ל 2, באמצעות 8 ביטים אפשר לייצג מספים בין 128- ל 127+.
- (3) `MOV BX,[AX]` → שגוי
 הסבר: אפשר להתייחס לכתובת בזיכרון בתוך [] (סוגרים מרובעים) רק עם האוגרים BX, DI, SI.
 אפשר להתייחס לכתובת בזיכרון בתוך [] (סוגרים מרובעים) רק עם האוגרים BX, DI, SI.
- (4) → שגוי
 הסבר: ההוראה CALL משנה את SP, דוחפת את כתובת החזרה למחסנית ולכן SP משתנה.
 ההוראה CALL לא משנה את SI
- (5) תקין
 ההוראה CALL מגדילה את ערכו של SP
 הסבר: ההוראה CALL מוציאה את המילה האחרונה מהמחסנית POP, ומשנה את ה-IP לכתובת שהוצאה מהמחסנית. הפקודה POP מגדילה את SP ב-2.
- (6) → שגוי
 הסבר: ההוראה RET משנה את ערכו של IP, לא תמיד דווקא מקטינה.
- (7) → שגוי
 הקטע הנ"ל יגרום ללולאה 4096 פעמים.
 CX מקבל את הערך 0 ויורד ל-1 ו-2 ו-3 וכו', עד שחוזר חזרה להיות 0 ומפסיק.
- (8) תקין
`XOR BX,BX` מאפסת את אוגר BX
- (9) תקין
`MOV CL,8`
`SHR DL,CL`
 הסבר: shr מבצע הזזה ציקלית ימינה ומכניס 0 מצד שמאל, אז 8 פעמים הזזה ציקלית והכנסת אפסים תאפס את אוגר DL
- (10) `NUM1 DB 100`
`NUM2 DB -156` → שגוי (מעל 128-)

ב.

```
MOV AL,-1
MOV DL,00001111B
MOV CL,4
AND AL,DL
SHR AL,CL
```

(1)

DX		CX		AX	
DH	DL	CH	CL	AH	AL
	0Fh		04h		0FFh
	0Fh				0Fh
					00h

הקטע מאפס את AL

(2) תקין MOV DL,00001111B

שגוי → MOV DL,00001111

הסבר: הפקודה בצורה כזו אומרת כי יש להכניס את המספר העשרוני 1111 לאוגר DL ואוגר זה בגודל בית יכול להכיל עד 255 בעשרוני.

עבודה 8

```
ARR DB 03H,08H,0CH,01H,04H, 09H,3H
LEN = $-ARR
```

```
LEA    SI,ARR ; array address
PUSH   SI
MOV    AX,LEN ; array length
PUSH   AX
CALL   FINDSECPND ; find big and small
POP    DI
POP    AX
MOV    [DI],AH ; mov small on big
LEA    SI,ARR
PUSH   SI
MOV    AX,LEN
PUSH   AX
CALL   FINDSECPND ; find big again (and small)
POP    DI
POP    AX
```

```
MOV     DL , AL ; print answer
MOV     AH , 2
INT     21H
```

```
FINDSECPND PROC
; find big and small
```

```
MOV     BP,SP
MOV     BX,[BP+4]
MOV     CX,[BP+2]
MOV     AL,[BX]
MOV     AH,[BX]
DEC     CX
SHOV:   INC     BX
        CMP     [BX],AL
        JL      MOVON
        MOV     AL,[BX]
        MOV     DI,BX
MOVON:  CMP     [BX],AH
        JG      CONT
        MOV     AH,[BX]
CONT:   LOOP    SHOV
        MOV     [BP+2],DI
        MOV     [BP+4],AX
        RET
ENDP
```

פרק ב'

מבוא לחקר ביצועים

עאלה 9

עאלה 10

עאלה 11

עאלה 12

פרק ב'

מודלים חישוביים

הפתרון לפרק זה נכתב ע"י רחל לודמר.

תרגיל 13

א. (i) השפה המתקבלת היא $\{ab\}$.

(ii) השפה המתקבלת היא $\{c^n ac^m a \mid n, m \geq 0\}$.

(iii) השפה המתקבלת היא $\{bbb\}$.

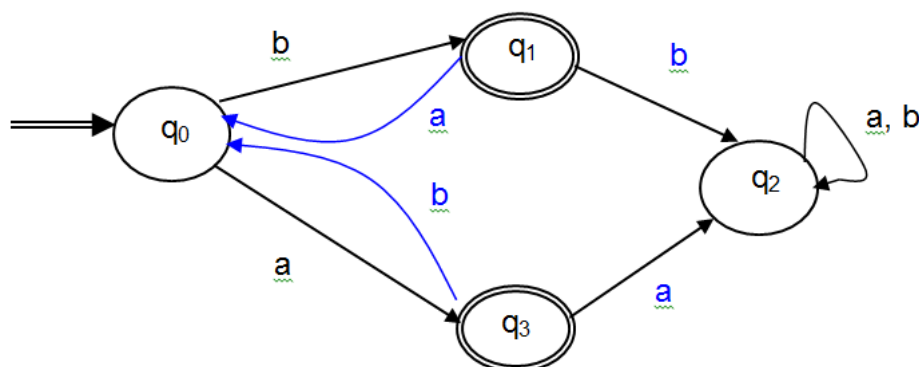
ב.

$$L_1: \rightarrow X \rightarrow Z \rightarrow Y \rightarrow R$$

$$L_2: \rightarrow Y \rightarrow X \rightarrow Z \rightarrow Z \rightarrow X \rightarrow R$$

$$L_3: \rightarrow Z \rightarrow Z \rightarrow Y \rightarrow X \rightarrow Z \rightarrow R$$

14 א



ב. שפה רגולרית ראשונה: $L_1 = \{a^n \mid n \geq 0\}$, ניתן לבנות לה אוטומט סופי דטרמיניסטי.

שפה רגולרית שנייה: $L_2 = \{b^n \mid n \geq 0\}$, ניתן לבנות לה אוטומט סופי דטרמיניסטי.

שרשורם $L_1 \cdot L_2 = \{a^n b^k \mid n, k \geq 0\}$ (עדיין השרשור הוא שפה רגולרית) אבל ב L_3 דורשים בנוסף לשרשור, שאורך המילים המשרשרות יהיה שווה.

$$L_3 = \{a^n b^k \mid n, k \geq 0, n = k\}$$

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

לכן נקבל את השפה:

וזו שפה אי רגולרית. בעצם אם ניקח כל שתי שפות רגולריות אינסופיות, ונשרשר אותם וגם נדרוש שאורך המילים המשרשרות יהיה זהה, התוצאה תהיה שפה אי רגולרית. (יש תלות של מניה).

$$L = (L_2 \cap \overline{L_3}) \cdot R(L_1)$$

$$\downarrow \quad \downarrow$$

$$\{c^n \mid n > 0\} \cdot \{b^k a^j \mid j, k \geq 0\}$$

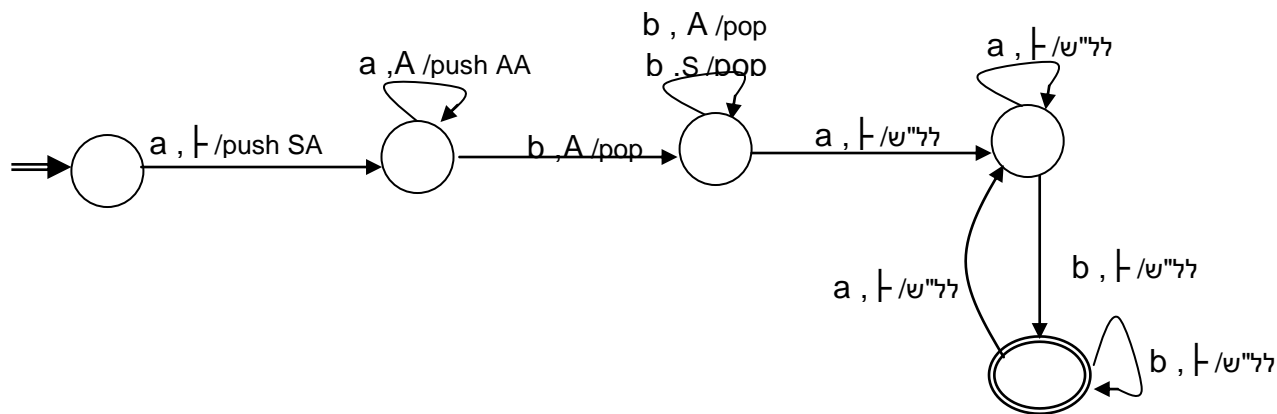
השפות L_1, L_2, L_3 הן רגולריות.

לכן, מתכונת סגירות משלים של שפה רגולרית הוא רגולרי, היפוך של שפה רגולרית, רגולרי. וכן מתכונת סגירות של חיתוך של שפות רגולריות הוא רגולרי, ושרשור של שפות רגולריות – רגולרי. נקבל שהשפה L רגולרית.

שאלה 15

א. המילה הקצרה היא: abbab.

ב.



שאלה 16

א.

$$w_1 = 0^i$$

$$w_2 = 0^j \mid j > i$$

$$w = 1^i$$

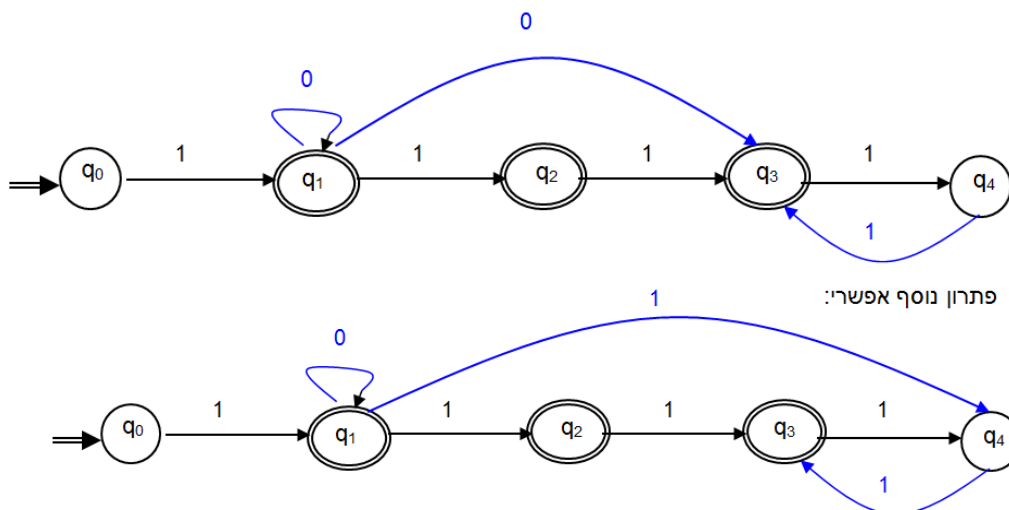
$$w_1 w = 0^i 1^i \mid i \leq i \leq 2i \Rightarrow w_1 w \in L_1$$

$$w_2 w = 0^j 1^i \mid j > i, \#_0(w_2 w) > \#_1(w_2 w) \Rightarrow w_2 w \notin L_1$$

הערה: הפתרון $w = 1^{j-1}$ אינו נכון.

מצד אחד מתקיים $j-1 \geq i$ אבל לא בהכרח מתקיים החסם השני $j-1 \leq 2i$.

ב.



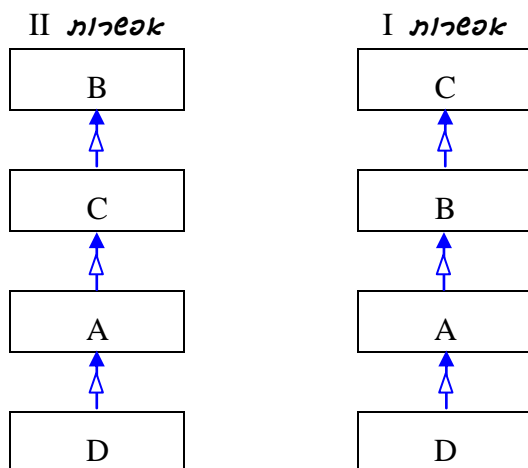
פרק ב'

תכנות מונחה עצמים Java / C#

הפתרון ל-3 השאלות הראשונות בתמ"ע זהה בשי השפות

תרגיל 17 / 21

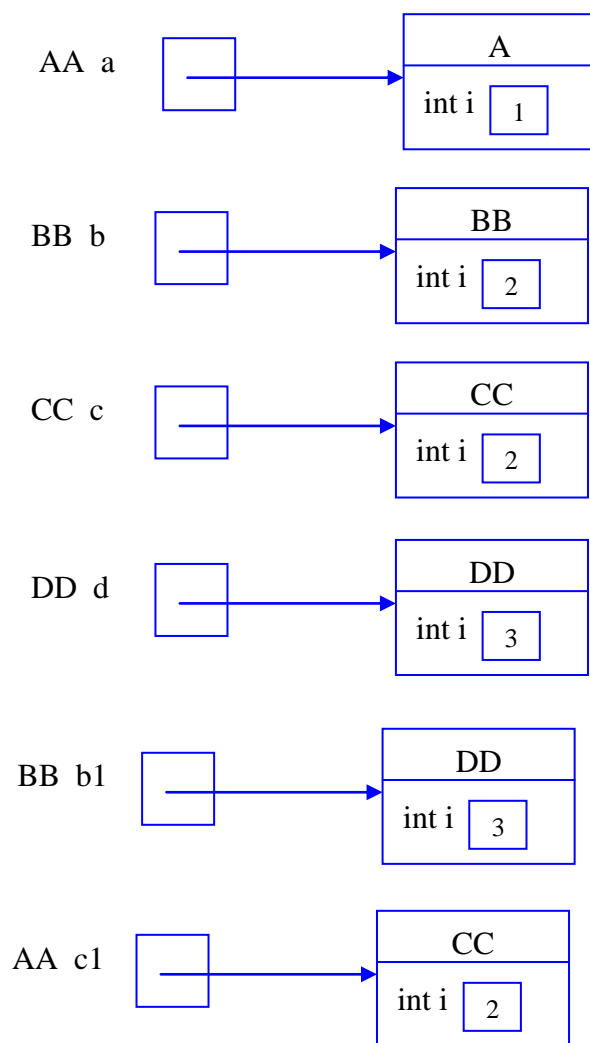
א. (1)



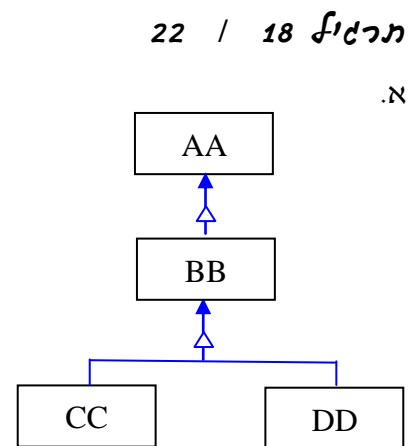
- (2) אפשרות I
ההוראות לא תקינות.
c (מטיפוס מחלקת העל) אינו מכיר את הפעולות שבמחלקה B (תת המחלקה) תקין.
אפשרות II
c יורש מהמחלקה B ולכן ירש גם את כל הפעולות שלה.
שני העצמים יפעילו את הפעולה המוגדרת ב-B

ב. AA לא יכול להיות ממשק כי לא ניתן ליצור עצם מסוג הממשק (ניתן ליצור עצם מסוג מחלקה המממשת את הממשק ולשים עליו הפנייה מטיפוס הממשק)

ג. נוצר עצם מסוג B עם הפנייה מסוג A למרות שבשתי המחלקות מוגדרת הפעולה f() והפעולה הבונה של A מפעילה את f(). הרי שמי שקובע איזו פעולה תופעל הוא טיפוס העצם ולא טיפוס ההפניה ולכן יופעל f() של המחלקה B.



ב.



א.

ג.

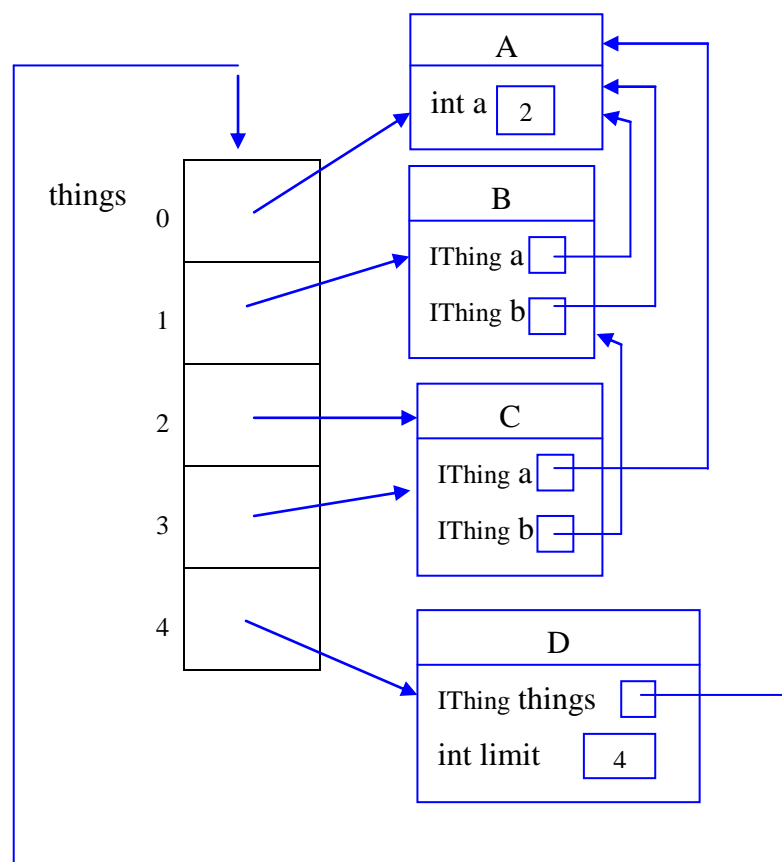
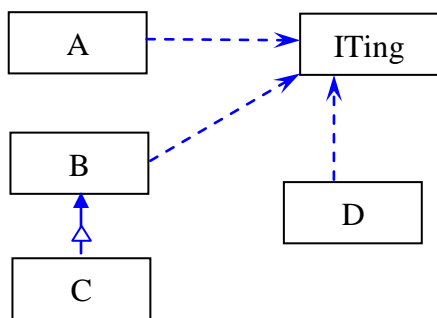
תקין. השמה חוקית של הפניה מסוג מחלקת העל לעצם מסוג תת המחלקה
 שגיאת קומפילציה. השמה **לא חוקית** של הפניה מסוג תת מחלקה לעצם מסוג מחלקת העל
 שגיאת זמן ריצה - שגיאת המרה. (AA עולם לא היה מסוג BB) (CC)b1;

ד.

ההוראה	הפלט	
b1.what(b)	false	ערכי i שונים
b1.what(b, 1)	true	
((CC)c1).what(c)	true	
d.what(a)	false	a אינו סוג של BB

תרגיל 19 / 23

א.



i	value ממחלקה	פלט
0	A	2
1	B	4
2	C	8
3	C	8
4	D	22

ב. עבור $\text{limit} = 5$ יפעיל את $\text{value}()$ שבמחלקה D בצורה רקורסיבית ללא תנאי עצירה, כלומר לולאה אינסופית.

הסיבה: הלולאה רצה עד $\text{limit} = 5$, כלומר, מנסה לסכם גם את value של $\text{things}[4]$ שהוא מסוג המחלקה D המכיל הפנייה למערך things של התכנית, ולכן יתחיל לסכם מחדש, וכך הלאה.

שאלה 20 - תמ"ע Java

```
public class Toy
{
    protected String name;           // שם הצעצוע
    protected double basePrice;       // מחיר בסיסי
    protected String color;           // צבע

    public Toy(String name, double basePrice, String color) {}
    //--- פעולה המחזירה את המחיר הבסיסי של הצעצוע ---
    public double price() {}
    //--- עדכון המחיר הבסיסי של הצעצוע ---
    public void updatePrice (double percent) {}
}
```

```
public class Doll extends Toy
{
    private int accNums;              // מספר אביזרים
    private double accPrice;          // מחיר אבזר בודד

    public Doll(String name, double basePrice, String color, int accNums, double accPrice) {}
    //--- פעולה המחזירה את מחיר הבובה כולל אבזרים ---
    public double price() {}
}
```

```
public class Car extends Toy
{
    private int size;                // גודל המכונית: 0 - קטנה, 1 - בינונית, 2 - גדולה

    public Car(String name, double basePrice, String color, int size) {}
    //--- פעולה המחזירה את מחיר המכונית בהתאם לגודלה ---
    public double price() {}
}
```

```
public static void main(String[] args)
{
    Doll d = new Doll ("Barbi", 100, "blond", 5, 10);
    Car c = new Car ("Bimba", 200, "red", 2);

    System.out.println("Doll price = " + d.price());
    System.out.println("Car price = " + c.price());
    System.out.println("both price = " + (d.price() + c.price()));
}
```

ג.

ב. הפעולה price() במחלקה Toy:

```
--- פעולה המחזירה את המחיר הבסיסי של הצעצוע ---  
public double price()  
{  
    return this.basePrice;  
}
```

הפעולה price() במחלקה Doll:

```
--- פעולה המחזירה את מחיר הבובה כולל אבזרים ---  
public double price()  
{  
    return super.price() + this.accNums * this.accPrice;  
}
```

הפעולה price() במחלקה Car:

```
--- פעולה המחזירה את מחיר המכונית בהתאם לגודלה ---  
public double price()  
{  
    double p = super.price();  
    if (this.size == 1)  
        p += 15;  
    if (this.size == 2)  
        p += 30;  
    return p;  
}
```

שאלה 24 - תמ"ע C# נכתב ע"י דיתה אוהב ציון

```
class Toy
{
    private string name;        // שם
    private double price;       // מחיר
    private string color;       // צבע
    public Toy(string name, double price, string color) {}
    public virtual double GetPrice() { return this.price; }
    public void UpdatePrice(int p) { }
}
```

```
class Doll: Toy
{
    private int AccNums;        // מספר אביזרים
    private double AccPrice;    // מחיר לאביזר
    public Doll (string name, double price, double AccPrice, int AccNums)
        : base(name, price, color)
    {
    }
    public override double GetPrice()
    {
        return base.GetPrice() + this.AccNums * this.AccPrice;
    }
}
```

```
class Car: Toy
{
    private int size;           // גודלים: 0,1,2
    public Car(string name, double price, string color, int size)
        : base(name, price, color) { }
    public override double GetPrice()
    {
        return base.GetPrice() + size * 15;
    }
}
```

```
Toy t1 = new Doll ("barbi", 65.60, "red", 12.5, 4);
Toy t2 = new Car ("bimba", 123, "green", 2);
Console.WriteLine(t1.GetPrice());
Console.WriteLine(t2.GetPrice());
```

ג.