

## מדעי המחשב

### פרק ראשון

#### שאלה 1

##### Java

```
//--- פעולה המקבלת שני מערכים באורך שונה של מספרים שלמים ---  
//--- ומחזירה מערך שלישי שאורכו כאורך המערך הארוך יותר ---  
//--- כך שחלקו הראשון הוא מכפלת האיברים במקומות המתאימים ---  
//--- בשני המערכים וחלקו השני הוא שאר איברי המערך הארוך יותר ---  
public static int[] multiply (int [] arr1, int [] arr2)  
{  
    //--- מיהו המערך הגדול יותר ---  
    int big, k, m;  
    if(arr1.length < arr2.length)  
    {  
        big = 2;  
        k = arr1.length;  
        m = arr2.length;  
    }  
    else  
    {  
        big = 1;  
        k = arr2.length;  
        m = arr1.length;  
    }  
  
    int[] arr3 = new int[m];  
  
    for (int i = 0 ; i < k ; i++)  
        arr3[i] = arr1[i] * arr2[i];  
  
    //--- העתקת ה"זנב" של המערך הארוך יותר ---  
    if (big == 1)  
    {  
        for (int i = k; i < m; i++)  
            arr3[i] = arr1[i];  
    }  
    else  
    {  
        for (int i = k; i < m; i++)  
            arr3[i] = arr2[i];  
    }  
    return arr3;  
}  
  
/*  
arr1 : 1 -4 4 9 2  
arr2 : 9 2 0 -1 3 11 23  
~~~~~  
multiply(arr1, arr2) : 9 -8 0 -9 6 11 23  
multiply(arr2, arr1) : 9 -8 0 -9 6 11 23  
*/
```

## C#

```

--- פעולה המקבלת שני מערכים באורך שונה של מספרים שלמים ---
--- ומחזירה מערך שלישי שאורכו כאורך המערך הארוך יותר ---
--- כך שחלקו הראשון הוא מכפלת האיברים במקומות המתאימים ---
--- בשני המערכים וחלקו השני הוא שאר איברי המערך הארוך יותר ---

```

0 references

```
public static int[] Multiply(int[] arr1, int[] arr2)
```

```
{
```

```
    --- מיהו המערך הגדול יותר ---
```

```
    int big, k, m;
```

```
    if (arr1.Length < arr2.Length)
```

```
    {
```

```
        big = 2;
```

```
        k = arr1.Length;
```

```
        m = arr2.Length;
```

```
    }
```

```
    else
```

```
    {
```

```
        big = 1;
```

```
        k = arr2.Length;
```

```
        m = arr1.Length;
```

```
    }
```

```
    int[] arr3 = new int[m];
```

```
    for (int i = 0; i < k; i++)
```

```
        arr3[i] = arr1[i] * arr2[i];
```

```
    --- העתקת ה"זנב" של המערך הארוך יותר ---
```

```
    if (big == 1)
```

```
    {
```

```
        for (int i = k; i < m; i++)
```

```
            arr3[i] = arr1[i];
```

```
    }
```

```
    else
```

```
    {
```

```
        for (int i = k; i < m; i++)
```

```
            arr3[i] = arr2[i];
```

```
    }
```

```
    return arr3;
```

```
}
```

```
    /*
```

```
    arr1 : 1 -4 4 9 2
```

```
    arr2 : 9 2 0 -1 3 11 23
```

```
    ~~~~~
```

```
    multiply(arr1, arr2) : 9 -8 0 -9 6 11 23
```

```
    multiply(arr2, arr1) : 9 -8 0 -9 6 11 23
```

```
    */
```

## שאלה 2

## Java

```
//---          מערך סריקה לפי ערכי תאים הוא מערך שבו
//--- קיים מסלול המתחיל בתא 0, ממשיך לתא שערכו כערך התא
//--- וסריקתו עוברת על כל התאים ומסתיימת בתא שערכו 0
public static boolean isPerfect (int[]arr)
{
    int n = arr.length;

    //--- יצירת מערך מונים ואיתחולו
    int [] arrT = new int[n];
    for (int i = 0 ; i < n ; i++)
        arrT[i] = 0;

    //--- מעבר על המסלול
    int k = arr[0];
    while (k != 0)
    {
        if (k < 0 || k >= n)
            return false;

        if(arrT[k] > 0)
            return false;    // כבר נתקלנו במספר זה

        arrT[k] ++;
        k = arr[k];
    }
    arrT[k] = 1;

    //--- בדיקה שערך כל התאים הוא 1
    for (int i = 0 ; i < arrT.length ; i++)
        if (arrT[i] == 0)
            return false;
    return true;
}
```

```
/*
arr1 : 2  2  3  2  0
is perfect(arr1): false

arr2 : 3  0  1  4  2
arrT  : 1  1  1  1  1
is perfect(arr2): true

arr3 : 1  2  5  1  3  4
is perfect(arr3): false

arr4 : 3  4  1  5  6  0  2
arrT  : 1  0  0  1  0  1  0
is perfect(arr4): false
*/
```

# C#

```
//--- מערך סריקה לפי ערכי תאים הוא מערך שבו ---
//--- קיים מסלול המתחיל בתא 0, ממשיך לתא שערכו כערך התא ---
//--- וסריקתו עוברת על כל התאים ומסתיימת בתא שערכו 0 ---
0 references
public static bool IsPerfect(int[] arr)
{
    int n = arr.Length;

    //--- יצירת מערך מונים ואיתחולו ---
    int[] arrT = new int[n];
    for (int i = 0; i < n; i++)
        arrT[i] = 0;

    //--- מעבר על המסלול ---
    int k = arr[0];
    while (k != 0)
    {
        if (k < 0 || k >= n)
            return false;

        if (arrT[k] > 0)
            return false; // כבר נתקלנו במספר זה

        arrT[k]++;
        k = arr[k];
    }
    arrT[k] = 1;

    //--- בדיקה שערך כל התאים הוא 1 ---
    for (int i = 0; i < arrT.Length; i++)
        if (arrT[i] == 0)
            return false;
    return true;
}
```

## שאלה 3

### Java

המחלקה Pixel

```
//--- סעיף א' ---
//--- פעולה המחזירה אמת אם הפיקסל אדום ושקר אחרת ---
//--- פיקסל אדום הוא פיקסל שבו כמות הצבע האדום ---
//--- בין 0 ו- 255 וכמות הצבע הירוק והכחול היא 0 ---
public boolean isRed()
{
    return this.red > 0 && this.red <= 255 &&
           this.green == 0 && this.blue == 0;
}

public boolean isBlue()
{...}

public boolean isGreen()
{...}

//--- סעיף ב' (2) ---
//--- פעולה המחזירה אמת אם צבע הפיקסל לבן ושקר אחרת ---
//--- פיקסל לבן הוא פיקסל שערך שלושת הצבעים שלו הוא 255 ---
public boolean isWhite()
{
    return this.red == 255 && this.green == 255 && this.blue == 255;
}

//--- סעיף ב' (2) ---
//--- פעולה המחזירה אמת אם צבע הפיקסל שחור ושקר אחרת ---
//--- פיקסל שחור הוא פיקסל שערך שלושת הצבעים שלו הוא 0 ---
public boolean isBlack()
{
    return this.red == 0 && this.green == 0 && this.blue == 0;
}
```

המחלקה Structure:

```
//--- (1) סעיף ב ---
//--- פעולה המחזיר אמת אם המערך מאוזן ושקר אחרת ---
//--- מערך מאוזן הוא מערך שבו מספר הנקודות האדומות ---
//--- זהה למספר הנקודות הכחולות ולמספר הנקודות הירוקות ---
//--- תיתכנה נקודות שאינן אדומות, ירוקות או כחולות ---
public boolean isBalanced()
{
    int countRed = 0, countGreen = 0, countBlue = 0;
    for(int i = 0 ; i < this.arr.length ; i++)
    {
        if (this.arr[i] != null && this.arr[i].isRed())
            countRed ++;
        else
            if (this.arr[i] != null && this.arr[i].isGreen())
                countGreen ++;
            else
                if (this.arr[i] != null && this.arr[i].isBlue())
                    countBlue ++;
    }
    return countRed == countGreen && countRed == countBlue;
}

//--- (2) סעיף ב ---
//--- פעולה המחזירה אמת אם המערך הוא מערך שחור-לבן ---
//--- מערך שחור-לבן הוא מערך שכל התאים בו שחורים או לבנים בלבד ---
public boolean isBlackWhite()
{
    int countBlack = 0, countWhite = 0;
    for(int i = 0 ; i < this.arr.length ; i++)
    {
        if (this.arr[i] != null && this.arr[i].isBlack())
            countBlack ++;
        else
            if (this.arr[i] != null && this.arr[i].isWhite())
                countWhite ++;
    }

    //--- האם יש לפחות אחד שחור ולפחות אחד לבן? ---
    if (countBlack == 0 || countWhite == 0)
        return false;

    //--- אם סכום המונים שווה לגודל המערך ---
    //--- אז כל הפיקסלים בו שחורים או לבנים בלבד ---
    return countBlack + countWhite == this.arr.length;
}
```

## C#

המחלקה Pixel:

```
//---          סעיף א          ---
//--- פעולה המחזירה אמת אם הפיקסל אדום ושקר אחרת ---
//--- פיקסל אדום הוא פיקסל שבו כמות הצבע האדום ---
//--- בין 0 ו- 255 וכמות הצבע הירוק והכחול היא 0 ---
1 reference
public bool IsRed()
{
    return this.red > 0 && this.red <= 255 &&
           this.green == 0 && this.blue == 0;
}

1 reference
public bool IsBlue()...

1 reference
public bool IsGreen()...

//---          סעיף ב' (2)          ---
//--- פעולה המחזירה אמת אם צבע הפיקסל לבן ושקר אחרת ---
//--- פיקסל לבן הוא פיקסל שערך שלושת הצבעים שלו הוא 255 ---
0 references
public bool IsWhite()
{
    return this.red == 255 && this.green == 255 && this.blue == 255;
}

//---          סעיף ב' (2)          ---
//--- פעולה המחזירה אמת אם צבע הפיקסל שחור ושקר אחרת ---
//--- פיקסל שחור הוא פיקסל שערך שלושת הצבעים שלו הוא 0 ---
0 references
public bool IsBlack()
{
    return this.red == 0 && this.green == 0 && this.blue == 0;
}
```

המחלקה Structure:

```
//--- (1) סעיף ב ---
//--- פעולה המחזיר אמת אם המערך מאוזן ושקר אחרת ---
//--- מערך מאוזן הוא מערך שבו מספר הנקודות האדומות ---
//--- זהה למספר הנקודות הכחולות ולמספר הנקודות הירוקות ---
//--- תיתכנה נקודות שאינן אדומות, ירוקות או כחולות ---
0 references
public bool IsBalanced()
{
    int countRed = 0, countGreen = 0, countBlue = 0;
    for (int i = 0; i < this.arr.Length; i++)
    {
        if (this.arr[i] != null && this.arr[i].IsRed())
            countRed++;
        else
            if (this.arr[i] != null && this.arr[i].IsGreen())
                countGreen++;
            else
                if (this.arr[i] != null && this.arr[i].IsBlue())
                    countBlue++;
    }
    return countRed == countGreen && countRed == countBlue;
}
```

```
//--- (2) סעיף ב ---
//--- פעולה המחזירה אמת אם המערך הוא מערך שחור-לבן ---
//--- מערך שחור-לבן הוא מערך שכל התאים בו שחורים או לבנים בלבד ---
0 references
public bool IsBlackWhite()
{
    int countBlack = 0, countWhite = 0;
    for (int i = 0; i < this.arr.Length; i++)
    {
        if (this.arr[i] != null && this.arr[i].IsBlack())
            countBlack++;
        else
            if (this.arr[i] != null && this.arr[i].IsWhite())
                countWhite++;
    }

    //--- האם יש לפחות אחד שחור ולפחות אחד לבן? ---
    if (countBlack == 0 || countWhite == 0)
        return false;

    //--- אם סכום המונים שווה לגודל המערך ---
    //--- אז כל הפיקסלים בו שחורים או לבנים בלבד ---
    return countBlack + countWhite == this.arr.Length;
}
```



## פרק שני

## שאלה 4

## Java

```

//--- פעולה המקבלת רשימת מספרים ורשימת תחומים ומחזירה אמת ---
//--- כל מספר ברשימת המספרים נמצא באחד התחומים ברשימת ---
//--- התחומים ושקר אחרת ---
//--- הערה, שתי הרשימות ממויינות בסדר עולה ---
public static boolean isIncluded (Node<Integer>l1, Node<Range>l2)
{
    int x;
    Range r;
    while (l1 != null && l2 != null)
    {
        r = l2.getValue();
        x = l1.getValue();

        if (x < r.getLow())
            return false; // x לא קיים ב- l2
        if (x > r.getHigh())
            l2 = l2.getNext(); // יש לעבור לתחום הבא
        else
            l1 = l1.getNext(); // נמצא בתחום הנוכחי
    }
    return true;
}

/*
l1: [-9, -8, -7, 12, 14, 15]
l2: [(-20, -10), (-9, 0), (2, 4), (12, 12), (14, 17)]
isIncluded(l1, l2) : true

l3: [1, 5, 8, 12]
l2: [(-20, -10), (-9, 0), (2, 4), (12, 12), (14, 17)]
isIncluded(l3, l2) : false
*/

```

## C#

```

//--- פעולה המקבלת רשימת מספרים ורשימת תחומים ומחזירה אמת ---
//--- כל מספר ברשימת המספרים נמצא באחד התחומים ברשימת ---
//--- התחומים ושקר אחרת ---
//--- הערה, שתי הרשימות ממייננות בסדר עולה ---
0 references
public static bool isIncluded(Node<int> lst1, Node<Range> lst2)
{
    int x;
    Range r;
    while (lst1 != null && lst2 != null)
    {
        r = lst2.GetValue();
        x = lst1.GetValue();

        if (x < r.GetLow())
            return false; // x לא קיים ב- lst2
        if (x > r.GetHigh())
            lst2 = lst2.GetNext(); // יש לעבור לתחום הבא
        else
            lst1 = lst1.GetNext(); // נמצא בתחום הנוכחי
    }
    return lst1 == null;
}

```

```

/*
lst1: [-9, -8, -7, 12, 14, 15]
lst2: [(-20, -10), (-9, 0), (2, 4), (12, 12), (14, 17)]
isIncluded(lst1, lst2) : true

lst3: [1, 5, 8, 12]
lst2: [(-20, -10), (-9, 0), (2, 4), (12, 12), (14, 17)]
isIncluded(lst3, lst2) : false
*/

```

## שאלה 5

### Java

```
//--- סעיף א' ---
//--- פעולה המקבלת מספר הקיים במחשנית sums ומחזירה ---
//--- ומחזירה מחשנית חדשה שבה כל האיברים מ-numbers שטכומס ---
//--- מספר זה. הפעולה הורסת את מחשנית numbers ---
public Stack<Integer> getNums (int x)
{
    Stack<Integer> sTemp = new Stack<Integer>();
    while (! this.numbers.isEmpty())
        sTemp.push(this.numbers.pop());

    Stack<Integer> stk = new Stack<Integer>();
    int sum = 0;
    while (sum < x)
    {
        this.numbers.push(sTemp.top());
        sum += sTemp.top();
        stk.push(sTemp.pop());
    }

    return stk;
}

//--- סעיף ב' ---
//--- הפעולה מוחקת את המספר ממחשנית numbers ומעדכנת את sums בהתאם ---
public void eraseNum (int x)
{
    //--- חיפוש x ומחיקת האיברים מ- sums ---
    Stack<Integer> sTemp = new Stack<Integer>();
    while (! this.numbers.isEmpty() && this.numbers.top() != x)
    {
        sTemp.push(this.numbers.pop());
        sums.pop();
    }

    //--- מחיקת x ותוספתו לסכומי משתי המחשניות ---
    this.numbers.pop();
    sums.pop();

    //--- החזרת האיברים תוך בניית sums ---
    int sum = 0;
    if (!this.sums.isEmpty())
        sum = this.sums.top();

    while (!sTemp.isEmpty())
    {
        this.numbers.push(sTemp.top());

        sum += sTemp.pop();
        this.sums.push(sum);
    }
}
```

## C#

```
//--- סעיף א' ---
//--- פעולה המקבלת מספר הקיים במחסנית sums ומחזירה ---
//--- ומחזירה מחסנית חדשה שבה כל האיברים מ- numbers שסכומם ---
//--- מספר זה. הפעולה הורסת את מחסנית numbers ---
0 references
public Stack<int> GetNums(int x)
{
    Stack<int> sTemp = new Stack<int>();
    while (!this.numbers.IsEmpty())
        sTemp.Push(this.numbers.Pop());

    Stack<int> stk = new Stack<int>();
    int sum = 0;
    while (sum < x)
    {
        this.numbers.Push(sTemp.Top());
        sum += sTemp.Top();
        stk.Push(sTemp.Pop());
    }

    return stk;
}

//--- סעיף ב' ---
//--- פעולה המוחקת מספר ממחסנית numbers ומעדכנת את מחסנית sums בהתאם ---
0 references
public void EraseNum(int x)
{
    //--- sums -ומחיקת האיברים מ x חיפוש ---
    Stack<int> sTemp = new Stack<int>();
    while (!this.numbers.IsEmpty() && this.numbers.Top() != x)
    {
        sTemp.Push(this.numbers.Pop());
        sums.Pop();
    }

    //--- ותוספתו לסכום משתי המחסניות x מחיקת ---
    this.numbers.Pop();
    sums.Pop();

    //--- sums החזרת האיברים תוך בניית ---
    int sum = 0;
    if (!this.sums.IsEmpty())
        sum = this.sums.Top();

    while (!sTemp.IsEmpty())
    {
        this.numbers.Push(sTemp.Top());

        sum += sTemp.Pop();
        this.sums.Push(sum);
    }
}
```

## שאלה 6

א. stackSod1 :

st בחזרה	val	st ריקה?	element	st בכניסה
[ 6, 3, 7, 4, 9 ]	6	לא	9	[ 6, 3, 7, 4 ]
[ 3, 7, 4, 9 ]	3	לא	9	[ 3, 7, 4 ]
[ 7, 4, 9 ]	7	לא	9	[ 7, 4 ]
[ 4, 9 ]	4	לא	9	[ 4 ]
[ 9 ]		כן	9	[ ]

המחסנית לאחר הזימון של stackSod1 : [ 6, 3, 7, 4, 9 ]

מטרת הפעולה : הפעולה דוחפת את element לתחתית המחסנית st

סיבוכיות הפעולה :  $O(n)$  כאשר n מייצג את מספר האיברים במחסנית.

הפעולה מרוקנת את המחסנית בזימון, ומחזירה את האיברים בחזרה

סה"כ  $2n + 1$  צעדים  $\Leftarrow O(n)$

ב. stackSod1 :

st בחזרה	אחרי StackSod1	val	st לא ריקה?	st בכניסה
[ 6, 3, 7, 4, 4, 7, 3, 6 ]	[ 3, 7, 4, 4, 7, 3, 6 ]	6	לא	[ 6, 3, 7, 4 ]
[ 3, 7, 4, 4, 7, 3 ]	[ 7, 4, 4, 7, 3 ]	3	לא	[ 3, 7, 4 ]
[ 7, 4, 4, 7 ]	[ 4, 4, 7 ]	7	לא	[ 7, 4 ]
[ 4, 4 ]	[ 4 ]	4	לא	[ 4 ]
			כן	[ ]

המחסנית לאחר הזימון של 2stackSod : [ 6, 3, 7, 4, 4, 7, 3, 6 ]

מטרת הפעולה : הפעולה דוחפת עותק של איברי המחסנית לתחתית המחסנית st

כך שמתקבלת מחסנית סימטרית (ההדפסה יוצרת פאלינדרום)

סיבוכיות הפעולה :  $O(n^2)$  כאשר n מייצג את מספר האיברים במחסנית.

הפעולה מבצעת n פעמים :

הפעולה stackSum1 -  $O(n)$

ולאחר מכן דוחפת איבר לראש המחסנית -  $O(1)$

סה"כ  $(n + 1) * n$  צעדים  $\Leftarrow O(n^2)$

## שאלה 7

### Java

```
//--- פעולה המחזירה אמת אם קיים בעץ מסלול המתחיל בשורה ---
//--- ומכיל את כל אותיות המילה הנתונה, ושקר אחרת ---
public static boolean wordFromRoot(BinNode<Character>tree, String str)
{
    if (str.equals("")) // if (str.length() == 0)
        return true;
    if (tree == null)
        return false;
    if (tree.getValue() != str.charAt(0))
        return false;

    str = eraseFirst(str); // הפעולה נתונה ואין צורך לממש אותה
    return wordFromRoot(tree.getLeft(), str) || wordFromRoot(tree.getRight(), str);
}

//--- פעולה המוחקת מהמחרוזת את התו הראשון ---
public static String eraseFirst (String str)
{...}
```

### C#

```
//--- פעולה המחזירה אמת אם קיים בעץ מסלול המתחיל בשורה ---
//--- ומכיל את כל אותיות המילה הנתונה, ושקר אחרת ---
2 references
public static bool WordFromRoot(BinNode<char> tree, string str)
{
    if (str == "")
        return true;
    if (tree == null)
        return false;
    if (tree.GetValue() != str[0])
        return false;

    str = EraseFirst(str); // הפעולה נתונה ואין צורך לממש אותה
    return WordFromRoot(tree.GetLeft(), str) || WordFromRoot(tree.GetRight(), str);
}

//--- פעולה המוחקת מהמחרוזת את התו הראשון ---
1 reference
public static string EraseFirst(string str){...}
```

## פרק feישי

### מערכות מחשב ואסמבלי

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

עלף 8

א. נתון קטע תוכנית בשפת אסמבלי.

```
MOV SI,30h
MOV CL,4
MOV CH,4
MOV BH,0
A1:
MOV AX,[SI]
MOV BL,AL
ROR BL,CL
DEC BL
MOV [SI],BX
ADD SI,2
DEC CH
CMP CH,0
JNE A1
```

AX		BX		CX		SI
AH	AL	BH	BL	CH	CL	
		00h		04h	04h	0030h
00h	25h		25h			
			52h			
			51h			
				03h		0032h
00h	3Ah					
			3Ah			
			0A3h			
			0A2h			
				02h		0034h
00h	7Bh					
			7Bh			
			0B7h			
			0B6h			
				01h		0036h
00h	93h		93h			
			39h			
			38h			
				00h		0038h

address	30h	31h	32h	33h	34h	35h	36h	37h
value	25h	00h	3Ah	00h	7Bh	00h	93h	00h
	51h	00h	0A2h	00h	0B6h	00h	38h	00h

קטע התוכנית עובר על מערך של מילים בגודל 4 החל מכתובת 30h, הופך את הבית התחתון בכל מילה (low) מחליף בין העשרות לאחדות ע"י Ror של 4 ביטים, ומהערך שהתקבל מוריד 1 את התוצאה כולה של מילה, מכניס חזרה למקום המתאים במערך. המערך הוא של מילים אבל הסיבוב וההפחתה היא רק לבית אחד מתוך כל מילה.

ב. להלן קטע תוכנית ה Java

```
if (x < y + z)
{
    do {
        if (y >= z || x < y)
            x += y;
        z--;
    } while (z > 0);
}
```

כתוב קטע מתאים באסמבלי.

המשתנים AX BX CX בהתאמה מתייחסים ל x y ו z מכוונים.

```
MOV AX, 12 ; X
MOV BX, 4 ; Y
MOV CX, 3 ; Z

MOV DX, BX
ADD DX, CX
CMP AX, DX
JGE SOF

AGAIN:
    CMP BX, CX
    JGE GOOD
    CMP AX, BX
    JGE NOTGOOD

GOOD:
    ADD AX, BX

NOTGOOD:
    DEC CX
    CMP CX, 0
    JG AGAIN

SOF:
```



## שאלה 9

"זוגות משני הצדדים" מערך זוגי יורד

ARR DB 45,43,40,17,16,16,20,12,25,12,1,11,32,30,36,40,43,32,34,43

LEN EQU \$-ARR

REZ DB ?

MOV REZ,0

MOV SI,0

MOV DI, LEN

DEC DI     ± DI=19

MOV CX, LEN

SHR CX,1     ± CX=10

MOV AL, ARR[SI]

ADD AL, ARR[DI]     ± AL KEEP LAST COUPLE

INC SI

DEC DI

DEC CX

AGAIN:

MOV DL, ARR[SI]

ADD DL, ARR[DI]

CMP DL, AL

JGE SOF

MOV AL,DL     ; KEEP LAST IN AL

INC SI

DEC DI

LOOP AGAIN

MOV REZ,1

SOF:

NOP

### אלגוריתמים

הפתרון לפרק זה נכתב ע"י:

**עאלה 10**

**עאלה 11**

## מודלים חישוביים

הפתרון לפרק זה נכתב ע"י: יקיר כהן

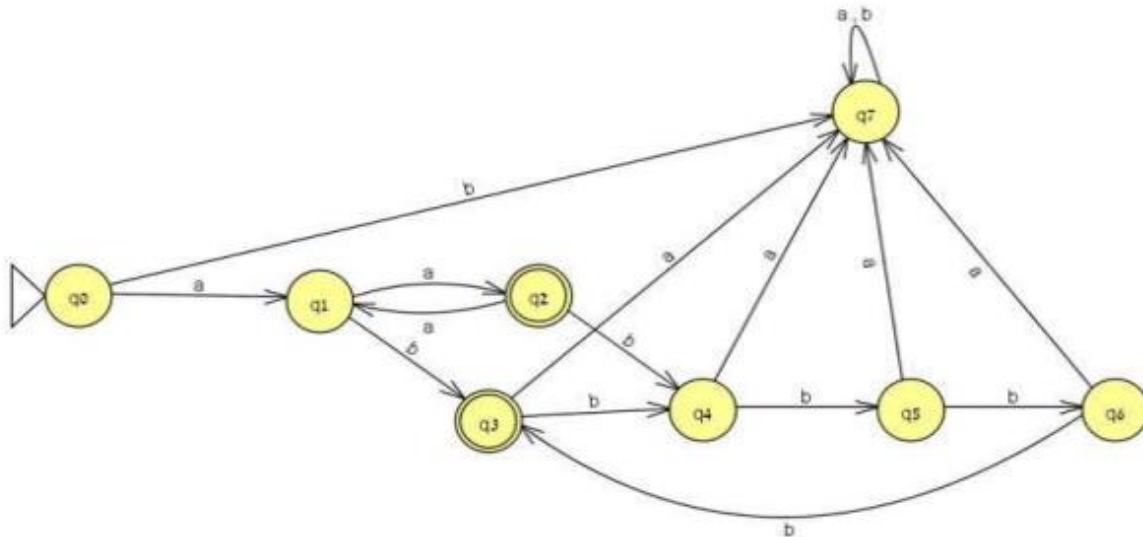
שאלה 12

$$L = \{a^n b^m \mid n > 0, m \geq 0, m \% 4 = m \% 2\}$$

א.

הסבר: $n = 1, m = 0$ - $1 \% 2 \neq 0 \% 4$	a - לא שייכת לשפה.
הסבר: $n = 1, m = 1$ - $1 \% 2 = 1 \% 4$	ab - שייכת לשפה.
הסבר: $n = 3, m = 3$ - $3 \% 2 \neq 3 \% 4$	aaabbbb - לא שייכת לשפה.
הסבר: $n = 3, m = 5$ - $3 \% 2 = 5 \% 4$	aaabbbbb - שייכת לשפה.
הסבר: $n = 2, m = 2$ - $2 \% 2 \neq 2 \% 4$	aabb - לא שייכת לשפה.
הסבר: $n = 3, m = 1$ - $3 \% 2 = 1 \% 4$	aaab - שייכת לשפה.
הסבר: $n = 2, m = 0$ - $2 \% 2 = 0 \% 4$	aa - שייכת לשפה.
הסבר: $n = 0, m = 4$ - לא עומד בתנאי $n > 0$	bbbb - לא שייכת לשפה.
הסבר: $n = 4, m = 0$ - $4 \% 2 = 0 \% 4$	aaaa - שייכת לשפה.
הסבר: $n = 1, m = 3$ - $1 \% 2 \neq 3 \% 4$	abbb - לא שייכת לשפה.

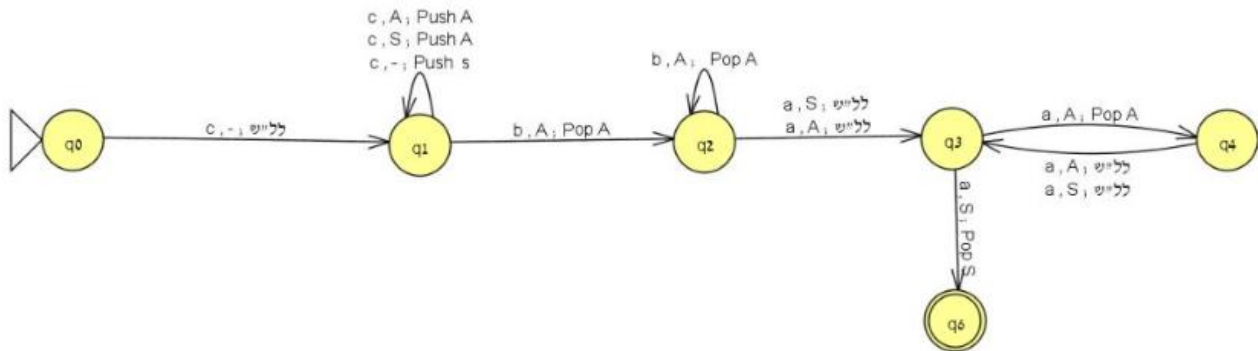
ב.



## שאלה 13

א.  $cccbaa$ : המילה הכי קצרה,  $n_{min} = 1, k_{min} = 1$  (1)

(2) אוטומט מחסנית המקבל את  $L_1$ :



ב.

$$L_1 = \{c^{1+k+n}b^ka^{2n} | n, k \geq 1\}$$

$$R(L_1) = \{a^{2n}b^kc^{1+k+n} | n, k \geq 1\}$$

$$L_2 = L_1 \cdot d \cdot R(L_1) = \{c^{1+k+n}b^ka^{2n}da^{2m}b^jc^{1+j+m} | n, k \geq 1, m, j \geq 1\}$$

הסבר:  $L_2$  ישרק d אחד בין  $L_1$  ל-  $R(L_1)$  ולא dd.

הסבר:  $cccbaa \notin R(L_1)$  (המילה מימין ל-d)

הסבר:  $cccbaa \in L_1$  (עבור  $n = 1, k = 1$ ),  
 $aaaabccccc \in R(L_1)$  (עבור  $m = 2, j = 1$ )  
ובין שתי המילים יש d.

cccbaaddaabc - לא שייכת לשפה.

cccbaadcccbaa - לא שייכת לשפה.

cccbaadaaaabccccc - שייכת לשפה.

תכנות מונחה עצמים java  
פתרון לפרק זה נכתב ע"י: אביטל Evi גרינוואלד

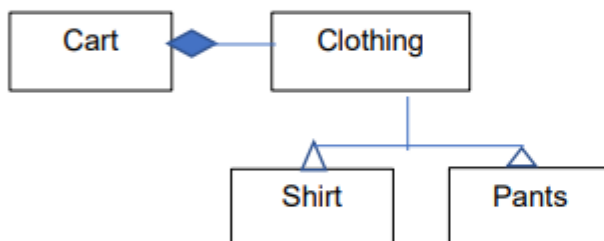
שאלה 14

השאלה עוסקת בירושה, העמסת פעולות, היררכיה בין מחלקות

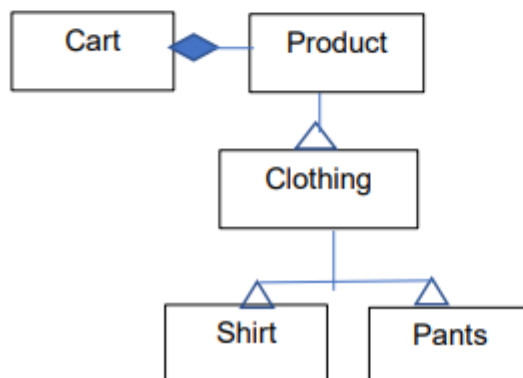
א-1 כותרת המחלקה Cart המייצגת סל קניות

<pre>public class Cart {     private String name;     private Clothing[] arr;     private int current;</pre>	<p>כותרת המחלקה</p> <p>שם הקונה</p> <p>מערך בגדים / חולצות ומכסניים</p> <p>מספר מוצרים נוכחי בקנייה</p>
--	---

א-2 תרשים היררכית המחלקות



ב-1 היררכית המחלקות לאחר השינוי



ב-2 כותרת המחלקה Product והתכונות

<pre>public class Product {     private int id;     private double price;</pre>	<p>כותרת המחלקה</p> <p>מספר זיהוי מוצר</p> <p>מחיר מוצר</p>
---	---

3-ב השינויים במחלקות לאחר הוספת המחלקה Product

הערה במחלקה Product יש פעולה בונה המתחלת מוצר עם הפרמטרים id, price

שם המחלקה	השינויים
Clothing	<pre>public class Clothing extends Product {     private String fabric;     private String color; }</pre>
Cart	<pre>public class Cart {     private String name;     private Product[] arr;     private int current; }</pre>

ג- הוספת פעולה של הוזלת מחיר

שם המחלקה	פעולת מחיר מוצר לאחר הנחה
Product	<pre>public double getDiscountPrice() {     return this.price * 0.9; }</pre>
Clothing	<pre>public double getDiscountPrice() {     return super.getDiscountPrice() - 10; }</pre>

## שאלה 15

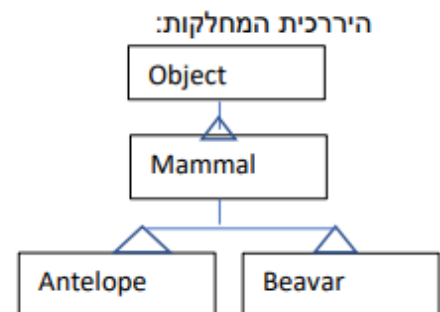
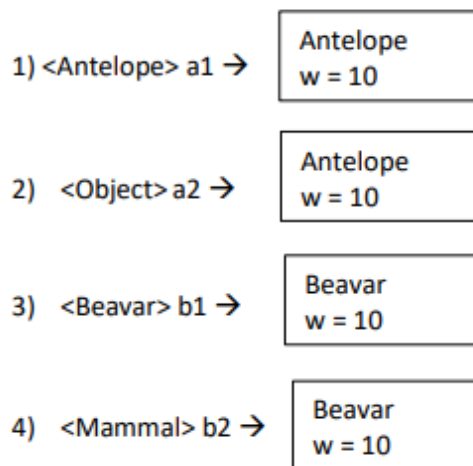
השאלה עוסקת בירושה, דריסת פעולות והעמסת פעולות, זיהוי שגיאות.

1. Antelope a1 = new Antelope(10);
2. Object a2 = new Antelope(10);
3. Beavar b1 = new Beavar(10);
4. Mammal b2 = new Beavar(10);

הפעולה הראשית במחלקה Program :

לפני שנענה על השאלה [ לא בקשו ]  
א- נצייר היררכית המחלקות.

ב- נצייר תרשים עצמים . [ מומלץ להוסיף סימון נקודת המבט על העצם ]



1	System.out.println(a1.weight);	שגיאת תחביר. אין אפשרות לגשת באופן ישיר אל התכונה weight מחוץ למחלקה Mammal או מחלקות הבת שלה. לציין באם המחלקה Program היתה כתובה באותה החבילה, בגלל שהתכונה wight מוגדרת כ protected , היה אפשר לגשת אליה באופן ישיר.
2	System.out.println(((Beavar)a2).getWeight());	שגיאת ריצה. העצם a2 נוצר מטיפוס Antelope , בזמן ריצה התוכנית תנסה להמיר אותו לטיפוס Beavar ותגלה שלא נוצר ממנו. המרות מתגלות בזמן ריצה.
3	System.out.println(a1.isSame(a2));	שגיאת תחביר. אין התאמה בטיפוסים המועברים בפרמטר אל הפעולה isSame a1 הוא מטיפוס Antelope קיימות עיבורו שתי פעולות isSame האחת שמקבלת עצם מטיפוס Mammal והשנייה עצם מטיפוס Antelope ואילו הפרמטר המועבר a2 הוא מטיפוס Object
4	System.out.println(a2.isSame(a1));	שגיאת תחביר. a2 הוא מטיפוס Object . למחלקה Object אין פעולה כזו isSame

5	<code>System.out.println(b1.isSame(b2));</code>	<b>תקין.</b> b1 מטיפוס Beavar ויש לו פעולה כזו עם אותה החתימה. יפעיל את הפעולה שכתובה ב Beavar . הפלט:  <b>In Beavar true</b>
6	<code>System.out.println(b2.isSame(b1));</code>	<b>תקין.</b> b2 גם הוא נוצר מטיפוס Beavar ולכן אותו הפלט  <b>In Beavar true</b>
7	<code>System.out.println(a1.isSame((Beavar) b2));</code>	<b>תקין.</b> a1 מטיפוס Antelope שיושם מ Mammal . הפרמטר שמקבל הוא מטיפוס Beavar שהוא Mammal ולכן מפעיל את הפעולה שמחלקת Antelope מקבלת בירושה מ Mammal ולכן הפלט הוא:  <b>InMammal false</b>
8	<code>System.out.println(a1.isSame((Antelope)a2));</code>	<b>תקין.</b> a1 מטיפוס Antelope והוא מקבלת כפרמטר עצם בנוצר מ Antelope ומומר אליו בחזרה מ Object ולכן מפעיל את הפעולה isSame שכתובה במחלקה Antelope. הפלט הוא:  <b>In Antelope true</b>
9	<code>System.out.println(b1.isSame((Antelope) a2));</code>	<b>תקין.</b> b1 מפעיל את הפעולה isSame על עצם מטיפוס Antelope שיושם מ Mammal והפלט הוא:  <b>In Beavar false</b>
10	<code>System.out.println(b1.isSame( (Beavar) a2));</code>	<b>שגיאת ריצה.</b> a2 נוצר מ Antelope. בזמן ריצה התוכנית תגלה שלא נוצר מ Beavar

אם הייתם נדרשים לתקן את שגיאות התחביר

		<b>הצעת תיקון</b>
1	<code>System.out.println(a1.weight);</code>	הפעלת <code>a1.getWeight()</code>
3	<code>System.out.println(a1.isSame(a2));</code>	המרת a2 בחזרה ל Antelope <code>System.out.println(a1.isSame((Antelope)a2));</code> ואז יהיה מתקבל פלט:  <b>In Antelope true</b>
4	<code>System.out.println(a2.isSame(a1));</code>	להמיר את a2 בחזרה מ Object ל antelope <code>System.out.println( ((Antelope)a2).isSame(a1));</code> במקרה כזה הפלט הוא:  <b>In Antelope true</b>

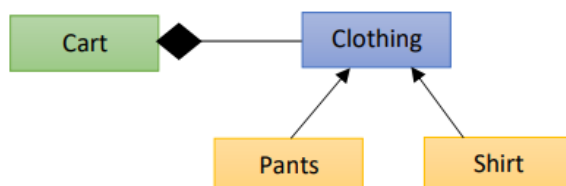


**תכנות מונחה עצמים C#**  
פתרון לפרק זה נכתב ע"י: דיתה אוהב ציון

**שאלה 16**

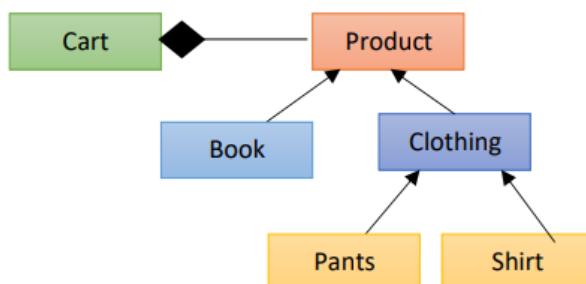
```
public class Cart
{
    private string name;
    private Clothing[] arr;
    private int current;
}
```

1.א. כותרות המחלקה Cart



2.א. תרשים מחלקות

לאחר השינוי



1.ב. תרשים מחלקות

```
public class Product
{
    private int id;
    private double price;
    public Product( int a, double b)
    { id = a; price = b; }
}
```

2.ב. המחלקה Product

ב.3 השינויים במחלקות הבאות (הבנאים רק להדגשת ההורשה)

```
public class Clothing:Product
{
    private string fabric;
    private string color;
    public Clothing(int a,string b,string c,double d)
        :base(a,d)
    {
        fabric = b; color = c;
    }
}
```

המחלקה Clothing

```
public class Book:Product
{
    private string bookName;
    private string author;
    public Book(int a, string b,string c,double d)
        :base(a,d)
    {
        bookName = b; author = c;
    }
}
```

המחלקה Book

```
public class Cart
{
    private string name;
    private Product[] arr;
    private int current;
}
```

המחלקה Cart

ג.

<pre>public virtual double GetDiscountPrice() {     return price*0.9; }</pre>	המחלקה Product
<pre>public override double GetDiscountPrice() {     return base.GetDiscountPrice()-10; }</pre>	המחלקה Clothing
<pre>public override double GetDiscountPrice() {     return base.GetDiscountPrice(); }</pre>	המחלקה Book (לא חובה)

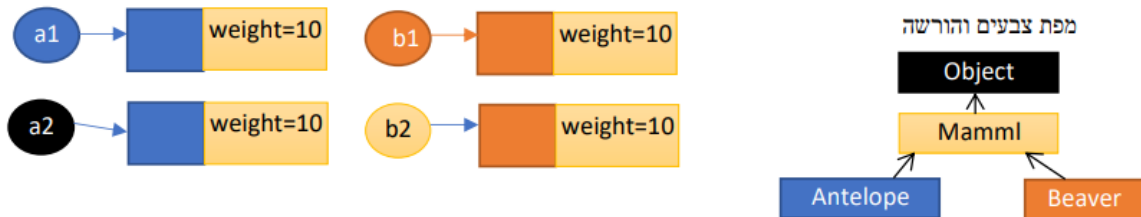
פעולות אלו מאפשרות לכל מוצר בטבלת הקניות להציג את המחיר שלו. (במחלקה ספר אין הנחה נוספת, לכן מחזיר את מחיר הבסיס)

## שאלה 17

הסבר: הפעולה IsSame במחלקה Antelope היא העמסה על הפעולה במחלקה Mammal (אותו שם, טיפוס פרמטר שונה)

הפעולה IsSame במחלקה Beaver היא דריסה (override) של גוף הפעולה במחלקה Mammal. (עצם מטיפוס Mammal המחזיק Beaver יפעיל את הפעולה IsSame של Beaver).

העצמים שנוצרו:



תיקון לא נדרש אם השורה שגויה.

Console.WriteLine(a1.weight);	1. שגוי. התכונה weight אינה מוכרת מחוץ למחלקה (תיקון: יש להשתמש בפעולה GetWeight()).
Console.WriteLine(((Beaver)a2).GetWeight());	2. שגוי. העצם המוחזק הוא Antelope ואי אפשר להמיר לטיפוס שאינו באותו מסלול הורשה.
Console.WriteLine(a1.IsSame(a2));	3. שגוי. הפעולה במחלקה Antelope מקבלת עצם מטיפוס Antelope ולא עצם מטיפוס Object. (אפשר לתקן ע"י המרה a1.IsSame((Antelope)a2))
Console.WriteLine(a2.IsSame(a1));	4. הפניה מטיפוס Object לא מכירה את הפעולות במחלקות היורשות ממנה. (תיקון ע"י המרה ((Antelope)a2).IsSame(a1))
Console.WriteLine(b1.IsSame(b2));	5. תיקון. הפלט: In Beaver true
Console.WriteLine(b2.IsSame(b1));	6. תיקון. הפלט: In Beaver true
Console.WriteLine(a1.IsSame((Beaver)b2));	7. תיקון. הפלט: In Mammal false
Console.WriteLine(a1.IsSame((Antelope)a2));	8. תיקון. הפלט: In Antelope true
Console.WriteLine(b1.IsSame((Antelope)a2));	9. תיקון. הפלט: In Beaver false
Console.WriteLine(b1.IsSame((Beaver)a2));	10. שגוי. אי אפשר להמיר לטיפוס Antelope לטיפוס Beaver (הם לא באותו מסלול הורשה).