

## מדעי המחשב ב'

### פתרון בחינת הקצרות

#### פרק א - עיצוב תכנה

#### שאלה 1:

א. טבלת מעקב למשפט הזימון: סוד3(a, 4, 5)

	1	2	3	4	5
a	4	3	7	6	10

ערך מוחזר	סוד2(a, n, p, n)	סוד1(a, n, 1, p)	n	p
4	4	3	5	4

טבלת מעקב ל-סוד1(a, 5, 1, 4)

n	p1	p	p = p1 ?	$ a[p] - a[p1] $ x	משפט זימון y	ערך מוחזר: $\max(x, y)$
5	1	4	F	2	סוד1(a, 5, 2, 4)	$\max(2, 3) \rightarrow 3$
5	2	4	F	3	סוד1(a, 5, 3, 4)	$\max(3, 1) \rightarrow 3$
5	3	4	F	1	סוד1(a, 5, 4, 4)	$\max(1, 0) \rightarrow 1$
5	4	4	T			0
ערך מוחזר: 3						

שימו ♥: במהלך הבחינה התפרסם עדכון לפיו גדול-בין-השניים (x,y) מקבל שני מספרים גדולים או שווים לאפס, ומחזיר את הגדול מביניהם.

טבלת מעקב ל-סוד 2 (a, 5, 4, 5)

n	p	p2	p = p2 ?	$ a[p] - a[p2] $ x	משפט זימון y	ערך מוחזר: $\max(x, y)$
5	4	5	F	4	סוד 2 (a, 5, 4, 4)	$\max(4, 0) \rightarrow 4$
5	4	4	T			0
ערך מוחזר: 4						

ב. סוד 3 מחזיר את ההפרש המקסימאלי בערך מוחלט, בין האיבר שבמקום p ושאר איברי המערך.

ג. סוד 4 (a, 5)

	1	2	3	4	5
a	4	3	7	6	10

n	k (a, 1, n) סוד 3	i	i < n	t (a, i, n) סוד 3	t > k	ערך מוחזר
5	6					
	7	2	T	7	T	
		3	T	4	F	
		4	T	4	F	
		5	F			7
ערך מוחזר: 7						

הערה לסעיף ג: נוצרה אי הבהירות בשורה (2) באלגוריתם סוד 4:

(2) עבור i מ-2 ועד n בצע

ניתן להריץ את הטבלה עבור  $i \leq n$  (כפי שמקובל בשפת פסקל) וניתן להריץ רק עד  $i < n$ .  
שני הפתרונות יתקבלו, שכן בשני המקרים אין הבדל בערך המוחזר.

ד. סוד 4 מחזיר את ההפרש המקסימלי, בערך מוחלט, בין הערך הגדול ביותר והערך הקטן ביותר במערך.

## שאלה 2:

א. ייצוג לטיפוס הנתונים **מיון בתחומים** - **sortByRange**

- מיון בתחומים = אובייקט / רשומה (פסקל) / מבנה (C):
- mins מערך חד ממדי בגודל 100 של מספרים שלמים.
  - values מערך חד ממדי בגודל 100 של רשימות של מספרים שלמים.
  - current מספר שלם לא שלילי.
  - size משתנה של המחלקה, ערכו 100, מייצג את גודל המערכים.

## פתרון בשפת Java:

```
public class SortByRange
{
    //--- תכונות המחלקה ---
    private int[] mins;
    private Node<Integer>[] values;
    private int current;
    public final int SIZE = 100;

    //--- בנאי - לא נדרש בבחינה ---
    public SortByRange ()
    {
        this.mins = new int[SIZE];
        this.values = new Node[SIZE]; // <Integer> ללא שימו לב!
        this.current = 0;
    }

    //--- סעיף ב' - הוספת קוליה לרשימה המתאימה כך שהמיון יישמר ---
    public void insert (int x)
    {
        //--- חיפוש הרשימה המתאימה ---
        int i = 0;
        while (i < this.current && this.mins[i] <= x)
            i++;
        //--- העצירה נעשית בתא הראשון שערכו גדול מהמספר החדש ---
        //--- לכן נחזור תא אחד ---
        i--;

        //--- הוספת האיבר בצורה ממויינת לרשימה המתאימה ---
        Node<Integer> pos = this.values[i];
        Node<Integer> prev = null;

        while (pos != null && pos.getValue() < x)
        {
            prev = pos;
            pos = pos.getNext();
        }
        if (prev == null)
            this.values[i] = new Node<Integer>(x, this.values[i]);
        else
            prev.setNext(new Node<Integer>(x, pos));
    }
}
```

```
//--- סעיף ג' - פעולה המוסיפה תחום חדש
//--- הנחה: התחום תקין ולא קיים במערך המינימום
public void addRange(int x)
{
    this.mins[this.current] = x;
    this.current++;
}
```

ד. (i) סיבוכיות הפעולה insert היא לינארית -  $O(n)$ :

חיפוש במערך min ב-m צעדים.

חיפוש המקום ברשימה values[i] ב-n צעדים.  $O(n) \leq O(m+n)$

(ii) סיבוכיות הפעולה addRange היא  $O(1)$ . גישה מיידית ל-current (הפעולה אינה תלויה בכמות הנתונים).

תכנית בדיקה: לא נדרשת בבחינה

```
public static void main(String[] args)
{
    SortByRange sbr = new SortByRange();
    sbr.addRange(x: 0);
    sbr.addRange(x: 101);
    sbr.addRange(x: 556);

    int[] arr = { 1, 700, 556, 120, 220, 9, 6, 99, 555 };
    for (int i = 0; i < arr.length; i++)
        sbr.insert(arr[i]);

    System.out.println("sbr: \n" + sbr.toString());
}

/*
sbr:
mins[0]: 0 ,      [1, 6, 9, 99]
mins[1]: 101 ,    [120, 220, 555]
mins[2]: 556 ,    [556, 700]
*/
```

פתרון בשפת C# :

```
class SortByRange                                     //--- סעיף א ---
{
    //--- תכונות המחלקה ---
    private int[] mins;
    private Node<int>[] values;
    private int current;
    public const int SIZE = 100;

    //--- בנאי - לא נדרש בבחינה ---
    1 reference
    public SortByRange ()
    {
        this.mins = new int[SIZE];
        this.values = new Node<int>[SIZE];
        this.current = 0;
    }

    //--- סעיף ב' - הוספת חוליה לרשימה המתאימה כך שהמיון יישמר
    1 reference
    public void Insert (int x)
    {
        //--- חיפוש הרשימה המתאימה ---
        int i = 0;
        while (i < this.current && this.mins[i] <= x)
            i++;
        //--- העצירה נעשית בתא הראשון שערכו גדול מהמספר החדש ---
        //--- לכן נחזור תא אחד ---
        i--;

        //--- הוספת האיבר בצורה ממויינת ---
        Node<int> pos = this.values[i];
        Node<int> prev = null;

        while (pos != null && pos.GetValue() < x)
        {
            prev = pos;
            pos = pos.GetNext();
        }
        if (prev == null)
            this.values[i] = new Node<int>(x, this.values[i]);
        else
            prev.SetNext(new Node<int>(x, pos));
    }
}
```

```
//--- סעיף ג' - פעולה המוסיפה תחום חדש
//--- הנחה: התחום תקין ולא קיים במערך המינימום
```

3 references

```
public void AddRange (int x)
{
    this.mins[this.current] = x;
    this.current++;
}
```

- ד. (i) סיבוכיות הפעולה היא לינארית  $O(n)$  במקרה הגרוע אנחנו נסרוק את כל הרשימה ובסוף נוסיף את האיבר sum, נניח שאורך הרשימה הארוכה ביותר היא  $n$  לכן הסיבוכיות היא לינארית.
- (ii)  $O(1)$  כל הפעולות הן קבועות.

תכנית בדיקה: לא נדרשת בבחינה

```
static void Main(string[] args)
{
    SortByRange sbr = new SortByRange();

    sbr.AddRange(0);
    sbr.AddRange(101);
    sbr.AddRange(556);

    int[] arr = { 1, 700, 556, 120, 220, 9, 6, 99, 555 };

    for (int i = 0; i < arr.Length; i++)
    {
        sbr.Insert(arr[i]);
    }

    Console.WriteLine("sbr: \n" + sbr.ToString());
}

/*
sbr:
mins[0]: 0,      [1, 6, 9, 99]
mins[1]: 101,    [120, 220, 555]
mins[2]: 556,    [556, 700]
*/
```

## פתרון בשפת Pascal:

**const**

size = 100;

**type**

arrType = **array** [1..size] **of integer**;

arrList = **array** [1..size] **of** list\_type; { list\_info\_type = integer; : list ביחידת הספרייה }

sortByRangeType = **record**

min : arrType;

values : arrList;

current : **integer**;

**end**;

ב. טענת כניסה : טנ"מ מיון-תחומים sbr, מספר שלם חיובי וגדול מהערך שב- min[0].  
 } טענת יציאה : num מוכנס לרשימה value המתאימה, על פי סדר המיון.

**procedure** insert (**var** sbr : sortByRangeType; num : **integer**);

**var**

i : **integer**;

**begin**

i = 0;

{--- חיפוש הרשימה המתאימה ---}

**while** (i < sbr.current - 1) **and** (sbr.min[i+1] < num) **do**

i := i + 1;

insertIntoSortedList (sbr.values[i], num);

**end**;

{ טענת כניסה : רשימה ממויינת של מספרים שלמים lst ומספר שלם וחיובי num  
 טענת יציאה : המספר num מוכנס למקומות המתאים ברשימה lst.  
 } הנחה : הרשימה L מאותחלת.

**procedure** insertIntoSortedList (**var** lst : arrList ; num : **integer**);

**var**

prev, pos : pos\_type;

x : **integer**;

found : **boolean**;

**begin**

prev := list\_anchore (lst);

pos := list\_next (lst, list\_anchore (lst));

found := **false**;

```

while (pos <> list_end(lst)) and not found do
  begin
    list_retrieve (lst, pos, x);
    if num < x then
      found := true
    else
      begin
        prev := pos;
        pos := list_next (lst, pos);
      end;
    end;
  list_insert (lst, prev, num);
end;

```

ג. טענת כניסה: טנ"מ מיון-תחומים sbr, מספר num שלם, חיובי וגדול מהערך המקסימלי { שב- values[current-1]. טענת יציאה: הוספת תחום חדש במערך min, ו- r הוא הערך שלו. הנחה: הטנ"מ sbr מאותחל, current < size }

```

procedure addRange (var sbr : sortByRangeType ; r : integer);
begin
  sbr.current := sbr.current + 1 ;
  sbr.min [sbr.current] := r ;
  list_init (sbr.values [sbr.current] );
end;

```

- ד. (i) סיבוכיות הפעולה insert היא לינארית -  $O(n)$ :  
 חיפוש במערך min ב- m צעדים.  
 חיפוש המקום ברשימה values[i] ב- n צעדים.  $O(n) \leq O(m+n)$
- (ii) סיבוכיות הפעולה addRange היא  $O(1)$ . גישה מיידית ל- current (הפעולה אינה תלויה בכמות הנתונים).



## פתרון בשפת C:

```

#define SIZE 100

typedef int arrType [SIZE] ;
typedef list_type arrList [SIZE] ;      // typedef int list_info_type; : list ביחידת הספרייה
typedef struct
{
    arrType min;
    arrList values;
    int current;
} sortByRangeType;

// טענת כניסה : טנ"מ מיון-תחומים sbr, num מספר שלם, חיובי וגדול מהערך שב- min[0].
// טענת יציאה : num מוכנס לרשימה value המתאימה, על פי סדר המיון.
void insert (sortByRangeType sbr , int num)
{
    int i = 0;
    //--- חיפוש הרשימה המתאימה ---
    while (i < sbr.current - 1 && sbr.min[i+1] < num)
        i ++ ;
    insertIntoSortedList (sbr.values[i], num);
}

// טענת כניסה : רשימה ממויינת של מספרים שלמים lst ומספר שלם וחיובי num
// טענת יציאה : המספר num מוכנס למקומות המתאים ברשימה lst.
// הנחה : הרשימה L מאותחלת
void insertIntoSortedList (list_type lst, int num)
{
    pos_type prev = list_anchor(lst);
    pos_type pos = list_next (lst, list_anchor(lst));
    while (pos != list_end(lst) && list_retrieve (lst, pos) < num)
    {
        prev = pos;
        pos = list_next (lst, pos);
    }
    insert (lst, prev, num);
}

```

ג. טענת כניסה: טנ"מ מיון-תחומים sbr, מספר num שלם, חיובי וגדול מהערך המקסימלי שב- values[current-1].  
 טענת יציאה: הוספת תחום חדש במערך min, ו- r הוא הערך שלו.  
 הנחה: הטנ"מ sbr מאותחל, current < size

```
void addRange (sortByRangeType sbr , int r)
{
    sbr.min [sbr.current] = r ;
    sbr.values [sbr.current] = list_init ();
    sbr.current ++ ;
}
```

- ד. (i) סיבוכיות הפעולה insert היא לינארית -  $O(n)$ :  
 חיפוש במערך min ב- m צעדים.  
 חיפוש המקום ברשימה values[i] ב- n צעדים.  $O(n) \leq O(m+n)$
- (ii) סיבוכיות הפעולה addRange היא  $O(1)$ . גישה מיידית ל- current (הפעולה אינה תלויה בכמות הנתונים).

### שאלה 3:

א. ייצוג לטיפוס הנתונים **משחק - Game** :

משחק = אובייקט / רשומה (פסקל) / מבנה (C):

- code מספר שלם המייצג את קוד המשחק.
- name מחרוזת המייצגת את שם המשחק.
- minAge הגיל המינימלי של שחקן שהמשחק מתאים לו.
- soldCopies מספר העותקים של המשחק שנמכרו בחנות.
- numPlayed מספר הפעמים הכולל ששיחקו במשחק.
- errStop מספר הפעמים הכולל שהמשחק נפסק בגלל תקלה.

**public class Game**

```
{
    private int code ;
    private String name;
    private int minAge;
    private int soldCopies;
    private int numPlayed ;
    private int errStop ;
    :
}
```

**פתרון בשפת Java:**

ייצוג לטיפוס הנתונים **מאגר-משחקים - GameData** : רשימה של משחקים

**public class GameData**

```
{
    private List <Game> g ; // רשימה של משחקים
    :
}
```

ואפשר גם מערך בגודל 100 של משחקים :  
**private Game [] g;**  
**private static int size = 100;**

ב.

מחלקה	שם הפעולה	תאור הפעולה
GameData	<b>void updateError (int code)</b>	פעולה המקבלת קוד של משחק code שנפסק בגלל תקלה, מאתרת אותו במאגר ומעדכנת את מספר השגיאות errStop במשחק זה.
GameData	<b>int soldMoreThenK (int k)</b>	פעולה המקבלת מספר שלם ולא שלילי k, סופרת ומחזירה את מספר המשחקים במאגר שנמכרו ביותר מ- k עותקים.

ג. (i) פעולה פנימית במחלקה :Game

// פעולה המחזירה את מספר הכוכבים במשחק הנוכחי.  
 // מספר הכוכבים מחושב כאחוז התקלות שנרשמו למשחק ביחס למספר המשחקים,  
 // ומציין את דירוג המשחק.

```
public int starNum ()
{
    double p = (double) this.errStop / this.numPlayed;
    if (p < 0.05)
        return 4;
    if (p < 0.25)
        return 3;
    if (p < 0.5)
        return 2;
    return 1;
}
```

(ii) פעולה פנימית במחלקה :GameData

// פעולה המקבלת גיל מינימלי minAge ומחזירה רשימה של כל הקודים של המשחקים  
 // המתאימים לשחקנים החל מגיל minAge ומדורגים כמשחקים של 4 כוכבים.

```
public List<Integer> gameList (int minAge)
{
    List<Integer> lst = new List<Integer> ();
    Game gm;
    Node <Integer> pos = this.getFirst();
    while (pos != null)
    {
        gm = pos.getInfo();
        if (gm.getMinAge() <= minAge && gm.starNum() == 4)
            lst.insert (null, gm.getCode()); // הרשימה נבנית בסדר הפוך
        pos = pos.getNext();
    }
    return lst;
}
```

```
public List<Integer> gameList (int minAge) // ובמימוש מערך:
{
    List<Integer> lst = new List<Integer> ();
    Game gm;
    for (int i = 0 ; i < this.g.length ; i++)
    {
        gm = this.g[i];
        if (gm.getMinAge() >= minAge && gm.starNum() == 4)
            lst.insert (null, gm.getCode()); // הרשימה נבנית בסדר הפוך
    }
    return lst;
}
```

## פתרון בשפת C#:

א. ייצוג לטיפוס הנתונים **משחק - Game** :

```
public class Game
{
    private int code ;
    private String name;
    private int minAge;
    private int soldCopies;
    private int numPlayed ;
    private int errStop ;
    :
}
```

ייצוג לטיפוס הנתונים **מאגר-משחקים - GameData** : רשימה של משחקים

```
public class GameData
{
    private List<Game> g ;      // רשימה של משחקים
    :
}
```

ואפשר גם מערך בגודל 100 של משחקים :  
 private Game [] g;  
 private static int size = 100;

ב.

מחלקה	שם הפעולה	תאור הפעולה
GameData	<b>void</b> UpdateError ( <b>int</b> code)	פעולה המקבלת קוד של משחק code שנפסק בגלל תקלה, מאתרת אותו במאגר ומעדכנת את מספר השגיאות errStop במשחק זה.
GameData	<b>int</b> SoldMoreThenK ( <b>int</b> k)	פעולה המקבלת מספר שלם ולא שלילי k, סופרת ומחזירה את מספר המשחקים במאגר שנמכרו ביותר מ- k עותקים.

ג. (i) פעולה פנימית במחלקה :Game

// פעולה המחזירה את מספר הכוכבים במשחק הנוכחי.  
 // מספר הכוכבים מחושב כאחוז התקלות שנרשמו למשחק ביחס למספר המשחקים,  
 // ומציין את דירוג המשחק.

```
public int StarNum ()
{
    double p = (double) this.errStop / this.numPlayed;
    if (p < 0.05)
        return 4;
    if (p < 0.25)
        return 3;
    if (p < 0.5)
        return 2;
    return 1;
}
```

(ii) פעולה פנימית במחלקה :GameData

// פעולה המקבלת גיל מינימלי minAge ומחזירה רשימה של כל הקודים של המשחקים  
 // המתאימים לשחקנים החל מגיל minAge ומדורגים כמשחקים של 4 כוכבים.

```
public List<int> GameList (int minAge)
{
    List<Integer> lst = new List<Integer> ();
    Game gm;
    Node <int> pos = this.GetFirst();
    while (pos != null){
        gm = pos.GetInfo();
        if (gm.GetMinAge() <= minAge && gm.starNum() == 4)
            lst.Insert (null, gm.GetCode()); // הרשימה נבנית בסדר הפוך
        pos = pos.GetNext();
    }
    return lst;
}
```

```
public List<int> GameList (int minAge) // ובמימוש מערך:
{
    List<int> lst = new List<int> ();
    Game gm;
    for (int i = 0 ; i < this.g.length ; i++)
    {
        gm = this.g[i];
        if (gm.GetMinAge() >= minAge && gm.starNum() == 4)
            lst.insert (null, gm.GetCode()); // הרשימה נבנית בסדר הפוך
    }
    return lst;
}
```

## פתרון בשפת Pascal:

א. ייצוג לטיפוס הנתונים **משחק - Game** :

**type**

```
Game = record
    code : integer;
    name : string;
    minAge : integer;
    soldCopies : integer;
    numPlayed : integer;
    errStop : integer;
end;
```

ייצוג לטיפוס הנתונים **מאגר-משחקים - GameData** : רשימה של משחקים

```
list_info_type = Game ;
GameData = list_type ;
```

הערה : ייצוג מאגר המשחקים ברשימה מחייב החזקת שתי יחידות ספרייה לרשימה.

```
list_info_type = Game      האחת רשימה שבה
list_info_type = integer   והשנייה רשימה שבה :
```

ואפשר לייצג גם במערך בגודל 100 של משחקים : `GameData = array [1..100] of Game ;`

ב.

שם הפעולה	תאור הפעולה
<b>procedure</b> updateError ( <b>var</b> G : gameData ; code : integer)	פעולה המקבלת את מאגר המשחקים G וקוד של משחק code שנפסק בגלל תקלה, מאתרת אותו במאגר ומעדכנת את מספר השגיאות errStop.
<b>function</b> SoldMoreThenK (G : gameData; k : integer) : integer;	פעולה המקבלת את מאגר המשחקים G, ומספר שלם ולא שלילי k, סופרת ומחזירה את מספר המשחקים במאגר שנמכרו ביותר מ- k עותקים.

ג. (i) פעולה המחזירה את מספר הכוכבים במשחק הנוכחי gm.  
מספר הכוכבים מחושב כאחוז התקלות שנרשמו למשחק ביחס למספר המשחקים, ומציין את דירוג המשחק. }

```
function StarNum (gm : Game) : integer ;
var
    p : real ;
begin
    p := gm.errStop / gm.numPlayed;
    if (p < 0.05) then
        StarNum := 4
    else
        if (p < 0.25) then
            StarNum := 3
        else
            if (p < 0.5) then
                StarNum := 2
            else
                StarNum := 1;
    end;
```

(ii) פעולה המקבלת את מאגר המשחקים G וגיל מינימלי minAge ומחזירה רשימה של כל הקודים של המשחקים המתאימים לשחקנים החל מגיל minAge ומדורגים כמשחקים של 4 כוכבים. הרשימה המוחזרת היא רשימה של integer. הנחה: המאגר G מאותחל. }

```
procedure GameList (var G : GameData ; minAge : integer ; lst : ;ist_type) ;
var
    gm : Game;
    pos : pos_type ;
begin
    list_init (lst);
    pos := list_next (G, list_anchor (G));
    while pos <> list_end (G) do
        begin
            list_retrieve (G, pos, gm);
            if (gm.minAge >= minAge ) and (starNum (gm) = 4) then
                insert (lst, list_anchor(lst), gm.code) ; { הרשימה נבנית בסדר הפוך }
            pos := list_next (G, pos);
        end;
    end;
```



ובייצוג מערך:

```
{
    פעולה המקבלת את מאגר המשחקים G וגיל מינימלי minAge
    ומחזירה רשימה של כל הקודים של המשחקים המתאימים לשחקנים
    החל מגיל minAge ומדורגים כמשחקים של 4 כוכבים.
    הרשימה המוחזרת היא רשימה של integer.
    הנחה: המאגר G מאותחל.
}

procedure GameList (var G : GameData ; minAge : integer ; lst : ;lst_type) ;
var
    gm : Game;
    i : integer;
begin
    list_init (lst);
    for i := 1 to 100 do
        begin
            gm := G[i];
            if (gm.minAge >= minAge ) and (starNum (gm) = 4) then
                insert (lst, list_anchor(lst), gm.code) ; { הרשימה נבנית בסדר הפוך }
            end;
        end;
    end;
end;
```

## פתרון בשפת C:

א. ייצוג לטיפוס הנתונים **משחק - Game** :

```
typedef struct
{
    int code ;
    char * name;
    int minAge;
    int soldCopies;
    int numPlayed ;
    int errStop ;
} Game;
```

ייצוג לטיפוס הנתונים **מאגר-משחקים - GameData** : רשימה של משחקים

```
typedef Game list_info_type ;           // טיפוס המידע ברשימה
typedef GameData list_type ;           // טיפוס מאגר המשחקים
```

הערה : ייצוג מאגר המשחקים ברשימה מחייב החזקת שתי יחידות ספרייה (מודולים) לרשימה.

```
typedef Game list_info_type ;           האחת רשימה שבה
typedef int list_info_type ;           והשנייה רשימה שבה :
```

ואפשר גם בעזרת מערך בגודל 100 של משחקים :  
**typedef Game GameData[100];**

ב.

שם הפעולה	תאור הפעולה
<b>void</b> updateError (GameData G , <b>int</b> code)	פעולה המקבלת את מאגר המשחקים G וקוד של משחק code שנפסק בגלל תקלה, מאתרת אותו במאגר ומעדכנת את מספר השגיאות errStop.
<b>int</b> soldMoreThenK (GameData G , <b>int</b> k)	פעולה המקבלת את מאגר המשחקים G, ומספר שלם ולא שלילי k, סופרת ומחזירה את מספר המשחקים במאגר שנמכרו ביותר מ- k עותקים.

ג. (i) פעולה המחזירה את מספר הכוכבים במשחק הנוכחי gm.  
 מספר הכוכבים מחושב כאחוז התקלות שנרשמו למשחק ביחס למספר המשחקים,  
 ומציין את דירוג המשחק.

```
int StarNum (Game gm)
{
    float p = (float) gm.errStop / gm.numPlayed;
    if (p < 0.05)
        return 4;
    if (p < 0.25)
        return 3;
    if (p < 0.5)
        return 2;
    return 1;
}
```

(ii) פעולה המקבלת את מאגר המשחקים G וגיל מינימלי minAge  
 ומחזירה רשימה של כל הקודים של המשחקים המתאימים לשחקנים  
 החל מגיל minAge ומדורגים כמשחקים של 4 כוכבים.  
 הרשימה המוחזרת היא רשימה של integer.  
 הנחה: המאגר G מאותחל.

```
list_type GameList (GameData G , int minAge)
{
    list_type lst = list_init ();
    Game gm;
    pos_type pos = list_next (G, list_anchor (G));
    while (pos != null)
    {
        gm = list_retrieve(G, pos);
        if (gm.minAge >= minAge && starNum(gm) == 4)
            list_insert (lst, list_anchor(lst), gm.code) ; // הרשימה נבנית בסדר הפוך
        pos = list_next(G, pos);
    }
    return lst;
}
```

ובמימוש מערך:

```
// פעולה המקבלת את מאגר המשחקים G וגיל מינימלי minAge
// ומחזירה רשימה של כל הקודים של המשחקים המתאימים לשחקנים
// החל מגיל minAge ומדורגים כמשחקים של 4 כוכבים.
// הרשימה המוחזרת היא רשימה של integer.
// הנחה: המאגר G מאותחל.

list_type GameList (GameData G , int minAge)
{
    list_type lst = list_init ();
    Game gm;
    for (int i = 0 ; i < 100 ; i ++)
    {
        gm = G[i] ;
        if (gm.minAge >= minAge && starNum(gm) == 4)
            list_insert (lst, list_anchor(lst), gm.code) ; // הרשימה נבנית בסדר הפוך
    }
    return lst;
}
```

#### שאלה 4 :

#### פתרון בשפת Java :

```
// טענת כניסה : מחסנית s שאיבריה הם מספרים שלמים.
// טענת יציאה : כל איבר-מחליף-כיוון ב-s מופיע פעמיים ברצף במחסנית.

public static void changeDirectionNum (Stack <Integer> s)
{
    Stack <Integer> sTemp = new Stack<Integer>();
    int first = 0, second = 0;
    boolean up;

    --- שליפת שני האיברים הראשונית מהמחסנית
    --- ובדיקה - האם הם מהווים תת-סדרה עולה (אמת) או יורדת (שקר)
    if ( ! s.isEmpty() )
    {
        first = s.pop();
        sTemp.push(first);
    }

    if ( ! s.isEmpty() )
        second = s.pop();
    if (first > second)
        up = false; // תת-הסדרה יורדת
    else
        up = true; // תת-הסדרה עולה

    --- בדיקת שאר איברי המחסנית
    first = second;
    sTemp.push (first);
    while ( ! s.isEmpty() )
    {
        second = s.pop();
        if ( (up && first > second) || (!up && second > first) )
        {
            sTemp.push (first); // דחיפת איבר מחליף כיוון נוסף
            up = ! up ; // החלפת כיוון תת-הסדרה
        }
        sTemp.push(second);
        first = second;
    }

    while ( ! sTemp.isEmpty() )
        s.push( sTemp.pop() );
}
```

- ב. סיבוכיות הפעולה היא  $O(n)$ .  
 עוברים על כל איברי המחסנית פעמיים - פעם אחת בלולאה הראשונה המחפשת את האיברים מחליפי הכיוון, ופעם שנייה בהחזרת האיברים בחזרה למחסנית.  
 כל פעולות המחסנית הן בסיבוכיות קבועה  $O(1)$

```
import java.util.Scanner;
public class Bagrut2008Q4
{
    /**
     *      2008 מספר 4 בבחינת בגרות
     *      איבר מחליף כיוון במחסנית
     *      נכתב ע"י: הילה קדמן
     */
    public static Scanner input = new Scanner(System.in);

    //      פעולה הקולטת איברים למחסנית
    static void stackFill(Stack<Integer> s)
    {
        System.out.print("type a number (-1 to finish) --> ");
        int num = input.nextInt();
        while (num != -1)
        {
            s.push(num);
            System.out.print("type a number (-1 to finish) --> ");
            num = input.nextInt();
        }
    }

    //      טענת כניסה: מחסנית s שאיבריה הם מספרים שלמים
    //      טענת יציאה: כל איבר מחליף כיוון ב-s
    //      מופיע פעמיים ברצף במחסנית
    public static void changeDirectionNum (Stack <Integer> s)
    {
        Stack <Integer> sTemp = new Stack<Integer>();
        int first = 0, second = 0;
        boolean up;

        //---      שליפת שני האיברים הראשונים מהמחסנית      ---
        //---      ובדיקה: האם הם מהווים תת-סדרה-עולה (אמת)      ---
        //---      או תת-סדרה-יורדת (שקר)      ---
        if ( ! s.isEmpty() )
        {
            first = s.pop();
            sTemp.push(first);          //      דחיפת איבר ראשון למחסנית
        }

        if ( ! s.isEmpty() )
            second = s.pop();

        if (first > second)
            up = false;          //      תת-סדרה יורדת
        else
            up = true;          //      תת-סדרה עולה

        //---      בדיקת שאר איברי המחסנית      ---
        first = second;
        sTemp.push(first);          //      דחיפת איבר שני למחסנית
        while (!s.isEmpty())
        {
            second = s.pop();
            if ((up && first > second) || (!up && second > first))
            {
                sTemp.push (first); //      דחיפת איבר מחליף כיוון נוסף
                up = ! up ;          //      החלפת כיוון הסדרה
            }
        }
    }
}
```

```

        sTemp.push(second);
        first = second;
    }

    //--- החזרת האיברים למחסנית ---
    while (! sTemp.isEmpty())
        s.push (sTemp.pop());
}

public static void main(String[] args)
{
    Stack <Integer> s = new Stack<Integer>();

    stackFill (s);
    System.out.println();

    System.out.println("before: " + s.toString());
    changeDirectionNum(s);
    System.out.println(s.toString());
}
}

/*

type a number (-1 to finish) --> 1
type a number (-1 to finish) --> 2
type a number (-1 to finish) --> 3
type a number (-1 to finish) --> 2
type a number (-1 to finish) --> 1
type a number (-1 to finish) --> 2
type a number (-1 to finish) --> -1

before: [2 , 1 , 2 , 3 , 2 , 1]
after: [2 , 1 , 1 , 2 , 3 , 3 , 2 , 1]

~~~~~

type a number (-1 to finish) --> 3
type a number (-1 to finish) --> 7
type a number (-1 to finish) --> 3
type a number (-1 to finish) --> 7
type a number (-1 to finish) --> 3
type a number (-1 to finish) --> -1

before: [3 , 7 , 3 , 7 , 3]
[3 , 7 , 7 , 3 , 3 , 7 , 7 , 3]

~~~~~

type a number (-1 to finish) --> 4
type a number (-1 to finish) --> 5
type a number (-1 to finish) --> 6
type a number (-1 to finish) --> 9
type a number (-1 to finish) --> -1

before: [9 , 6 , 5 , 4]
[9 , 6 , 5 , 4]

*/

```

פתרון נוסף בשפת Java: נכתב ע"י יבגני קנל

```
import java.util.Scanner;
import unit4.collectionsLib.*;

// 899205-2008 - Evgeny Kanel
class test4{
    static Scanner in=new Scanner(System.in);

    public static void fill(Stack<Integer> s)
    {
        int x = in.nextInt();
        while(x > 0)
        {
            s.push(x);
            x = in.nextInt();
        }
    }

    public static void test4a(Stack<Integer> s)
    {
        Stack<Integer> s1=new Stack<Integer>();
        int x = s.pop();
        s1.push (x);
        while (! s.isEmpty())
        {
            int y = s.pop();
            s1.push (y);
            if (! s.isEmpty())
            {
                int z = s.top();
                if(x<y && z<y || x>y && z>y)
                    s1.push (y);
                x = y;
            }
        }
        while (!s1.isEmpty())
            s.push (s1.pop());
    }

    public static void main(String[] args)
    {
        Stack<Integer> s=new Stack<Integer>();
        fill (s);
        System.out.println(s);
        test4a(s);
    }
}
```



פתרון בשפת C# - נכתב על ידי ראמי ג'באלי :

```
public static void NumChangeDirection(Stack<int> s)
{
    Stack<int> s2 = new Stack<int>();
    int a = 0, b = 0, c;
    if (!s.IsEmpty())
    {
        a = s.Pop();
        s2.Push(a);
    }
    if (!s.IsEmpty())
    {
        b = s.Pop();
        s2.Push(b);
    }
    while (!s.IsEmpty())
    {
        c = s.Pop();
        if (a>b && b<c || a<b && b>c)
        {
            s2.Push(b);
        }
        s2.Push(c);
        a = b;
        b = c;
    }
    while (!s2.IsEmpty())
        s.Push(s2.Pop());
}
```

ב. סיבוכיות כל פעולה במחלקה Stack היא  $O(1)$ , הפעולה עוברת פעם אחת על איברי המחסנית ובמהלך הפעולה מתבצעת פעולות השוואה שסיבוכיותו קבועה מכאן סיבוכיות הפעולה היא  $O(n)$ .

פלט:

הערה: אפס משמש לסיום הקלט:

```

C:\Windows\system32\cmd.exe
1
2
3
2
1
2
0
The Stack Befor Change:
[2,1,2,3,2,1]
The Stack After Change:
[2,1,1,2,3,3,2,1]
Press any key to continue . . .

```

```

C:\Windows\system32\cmd.exe
10
8
6
5
4
7
8
9
10
1
2
9
3
0
The Stack Befor Change:
[3,9,2,1,10,9,8,7,4,5,6,8,10]
The Stack After Change:
[3,9,9,2,1,1,10,10,9,8,7,4,4,5,6,8,10]
Press any key to continue . . .

```

**פתרון בשפת Pascal :**

```
{
    טענת כניסה : מחסנית s שאיבריה הם מספרים שלמים.
    טענת יציאה : כל איבר-מחליף-כיוון ב-s מופיע פעמיים ברצף במחסנית.
}

procedure changeDirectionNum (var s : Stack_type)
var    sTemp : stack_type;
      first, second : integer;
      up : boolean;
begin
    stack_init (sTemp);
    first := 0;      second := 0;
    {--- שליפת שני האיברים הראשונים מהמחסנית ---}
    {--- ובדיקה - האם הם מהווים תת-סדרה עולה (אמת) או יורדת (שקר) ---}
    if not stack_is_empty(s) then
        begin
            stack_pop (first);
            stack_push (sTemp, first);
        end;

    if not stack_is_empty(s) then stack_pop (second);
    if (first > second) then
        up := false;    { תת-הסדרה יורדת }
    else up := true;    { תת-הסדרה עולה }
    first := second;    {--- בדיקת שאר איברי המחסנית ---}
    stack_push (sTemp, first);
    while not stack_is_empty(s) do
        begin
            stack_pop (second);
            if (up and (first > second)) or (not up and (second > first)) then
                begin
                    stack_push (sTemp, first);    // דחיפת איבר מחליף כיוון נוסף
                    up := not up;                  // החלפת כיוון תת-הסדרה
                end;
            stack_push (sTemp, second);
            first = second;
        end;
    while not stack_is_empty(s) do
        begin
            stack_pop (second);
            stack_push (sTemp, first);
        end;
end;
```

- ב. סיבוכיות הפעולה היא  $O(n)$ .  
 עוברים על כל איברי המחסנית פעמיים - פעם אחת בלולאה הראשונה המחפשת את האיברים מחליפי הכיוון, ופעם שנייה בהחזרת האיברים בחזרה למחסנית.  
 כל פעולות המחסנית הן בסיבוכיות קבועה  $O(1)$

## פתרון בשפת C :

```
// טענת כניסה : מחסנית s שאיבריה הם מספרים שלמים.
// טענת יציאה : כל איבר-מחליף-כיוון ב-s מופיע פעמיים ברצף במחסנית.

void changeDirectionNum (stack_type *s)
{
    stack_type sTemp = stack_init ();
    int first = 0, second = 0;
    int up;

    //--- שלפת שני האיברים הראשונית מהמחסנית
    //--- ובדיקה - האם הם מהווים תת-סדרה עולה (1) או יורדת (-1)
    if ( ! stack_is_empty(*s) )
    {
        first = stack_pop (s);
        stack_push (sTemp, first);
    }

    if ( ! stack_is_empty(s) ) second = stack_pop (s);
    if (first > second)
        up = -1; // תת-הסדרה יורדת
    else
        up = 1; // תת-הסדרה עולה

    //--- בדיקת שאר איברי המחסנית
    first = second;
    stack_push (sTemp, first);
    while ( ! stack_is_empty(*s) )
    {
        second = stack_pop (s);
        if ( (up == 1 && first > second) || (up == -1 && second > first) )
        {
            stack_push (sTemp, first); // דחיפת איבר מחליף כיוון נוסף
            up = up * (-1); // החלפת כיוון תת-הסדרה
        }
        stack_push (sTemp, second);
        first = second;
    }

    while ( ! sTemp.isEmpty() )
        stack_push (s, stack_pop (sTemp));
}
```

- ב. סיבוכיות הפעולה היא  $O(n)$ .  
 עוברים על כל איברי המחסנית פעמיים - פעם אחת בלולאה הראשונה המחפשת את האיברים מחליפי הכיוון, ופעם שנייה בהחזרת האיברים בחזרה למחסנית.  
 כל פעולות המחסנית הן בסיבוכיות קבועה  $O(1)$

פרק ב'

מערכות מחשב ואסמבלר

הפתרון לפרק זה נכתב ע"י חוסיין זועבי

תרגיל 5:

- א. *i* מבצע את הנדרש.  
מוצא את הערכים המוחלטים ומשווה ביניהם. בסוף מאחסן באוגר AX את המספר שהערך המוחלט שלו הוא הגדול ביותר מבין שניהם.
- ii* אינו מבצע את הנדרש.  
ב- AX יאוחסן המספר שהערך המוחלט שלו הוא הקטן ביותר מבין שניהם.
- iii* אינו מבצע את הנדרש.  
כאשר הערך המוחלט של AX יותר גדול מהערך המוחלט של BX יאוחסן באוגר AX את ערכו המוחלט ולא את ערכו המקורי כפי שנדרש.

ב.

AX [H]	BX [H]	ZF	SF	CF
C83B	????	?	?	?
	A89C	?	?	?
9076		0	1	1
9077		0	1	0
	A863	0	1	0
38DA	A863	0	0	1

תרגיל 6:

א.

- i* XOR SI, SI  
*ii* DEC CX  
*iii* A2: POP AX  
*iv* JE A3  
*v* LOOP A2

- א. *i* אינו מבצע את הנדרש. מתבצע איפוס של האיבר שהמציין שלו הוא  $4*Y$ .  
*ii* אינו מבצע את הנדרש. ההוראה MOV [BX+ 3\*Y], AL אינה חוקית.  
*iii* מבצע את הנדרש. תוצאת הכפל  $3*Y$  תאוחסן ב- AX ומשם היא תועבר לאוגר האינדקס SI ואח"כ יאופס האיבר שהמציין שלו הוא  $3*Y$ .  
*iv* אינו מבצע את הנדרש. מתבצע איפוס של האיבר שהמציין שלו הוא Y.

תרגיל 7:

דרך א:

```
.8086
.model small
.stack 100h
.data
    a db 1,1,0,1,1,1,0,0
    b db 1,1,1,1,1,1,1,0
    c_ db 1,0,0,1,0,0,0,0
    d db 10 dup(?)
.code
start: mov ax,@data
        mov ds, ax
        mov cx,8
        mov si,7
        cld
again:  mov al, a[si]
        adc al, b[si]
        test al,2
        jz con
        sub al, 2
        stc
con:    mov d[si+2], al
        dec si
        loop again
        mov d[1],0
        adc d[1], 0
        mov cx,8
        mov si,7
        cld
again1: mov al, c_[si]
        adc d[si+2],al
        test d[si+2],2
        jz con1
        sub d[si+2],2
        stc
con1:   dec si
        loop again1
        adc d[1], 0
        test d[1],2
        jz con2
        sub d[1],2
        stc
con2:   mov d[0],0
        adc d[0], 0
        mov ah, 4ch
        int 21h
end start
```

```

start:  mov ax,@data                                דרך ב:
        mov ds, ax
        lea si,a+7
        push si
        call ad_bin
        mov bl, al
        lea si,b+7
        push si
        call ad_bin
        mov bh, al
        lea si,c_+7
        push si
        call ad_bin
        mov dl, al
        add bl, bh
        mov d[1], 0
        adc d[1], 0
        add bl, dl
        adc d[1], 0
        mov bh, d[1]
        and d[1], 01
        shr bh, 1
        mov d[0], bh
        mov si, 9
        mov cx, 8
con:    mov dl, bl
        and dl, 01
        mov d[si], dl
        shr bl, 1
        dec si
        loop con
        mov ah, 4ch
        int 21h

ad_bin proc
        mov bp, sp
        mov si, [bp+2]
        mov al, 0
        mov cl, 1
        mov ah, [si]
A1:     or al, ah
        dec si
        mov ah, [si]
        shl ah, cl
        inc cl
        cmp cl, 9
        jne A1
        ret 2
ad_bin endp
end start

```

דרך ג:

```

start:  mov ax,@data
        mov ds, ax
        mov cx,8
        mov si,7
        mov ah, 0
A1:     mov al, a[si]
        mov bl, al
        mov dl, al
        xor al, b[si]
        xor al,ah
        and bl, b[si]
        or  dl, b[si]
        and ah, dl
        or  ah, bl
        mov d[si+2], al
        dec si
        loop A1
        mov d[1],ah
        mov cx,8
        mov si,7
        mov ah,0
A2:     mov al, d[si+2]
        mov bl, al
        mov dl, al
        xor al, c_[si]
        xor al,ah
        and bl, c_[si]
        or  dl, c_[si]
        and ah, dl
        or  ah, bl
        mov d[si+2], al
        dec si
        loop A2
        add d[1], ah
        mov  dh, d[1]
        and d[1], 01
        shr dh,1

        mov d[0],dh

        mov ah, 4ch
        int 21h

end start
    
```



דרך ד:

```

start:  mov ax, @data
        mov ds, ax
        lea si, a+7
        lea di, b+7
        push si
        push di
        call ad_bin
        mov d[1], ah
        lea si, d+9
        lea di, c_+7
        push si
        push di
        call ad_bin
        add d[1], ah
        mov ah, d[1]
        and d[1], 01
        shr ah, 1
        mov d[0], ah
        mov ah, 4ch
        int 21h

ad_bin proc
        mov bp, sp
        mov si, [bp+4]
        mov di, [bp+2]
        mov cx, 8
        mov bx, 9
        mov ah, 0 ; mov CF=0 to ah
A1:     mov al, [si]
        mov dh, al
        mov dl, al
        xor al, [di]
        xor al, ah
        and dh, [di]
        or dl, [di]
        and ah, dl
        or ah, dh
        mov d[bx], al
        dec si
        dec di
        dec bx
        loop A1
        ret 4
ad_bin endp
end start
    
```

שאלה 8 :

```

XOR BL, BL
    MOV CX, 13
AG:  MOV DX, AX
    AND DL, 0FH
    CMP DL, 0BH
    JNE CON
    INC BL
CON: SHR AX, 1
    LOOP AG
    
```

פרק ב'

תורת המחשב

שאלה 9: הפתרון לשאלה 9 נכתב ע"י יבגני קנל

$$x + y - 1 = 1$$

$$2x + 3y + \alpha z = 3$$

$$x + \alpha y + 3z = 2$$

$$\left( \begin{array}{ccc|c} 1 & 1 & -1 & 1 \\ 2 & 3 & \alpha & 3 \\ 1 & \alpha & 3 & 2 \end{array} \right) \Rightarrow \left( \begin{array}{ccc|c} 1 & 1 & -1 & 1 \\ 0 & 1 & (\alpha+2) & 1 \\ 0 & (\alpha-1) & 4 & 1 \end{array} \right) \Rightarrow$$

$$\left( \begin{array}{ccc|c} 1 & 0 & -(\alpha+3) & 0 \\ 0 & 1 & (\alpha+2) & 1 \\ 0 & 0 & 4-(\alpha+2)(\alpha-1) & 1-(\alpha-1) \end{array} \right) \Rightarrow \left( \begin{array}{ccc|c} 1 & 0 & -(\alpha+3) & 0 \\ 0 & 1 & (\alpha+2) & 1 \\ 0 & 0 & -(\alpha-2)(\alpha+3) & (2-\alpha) \end{array} \right) \Rightarrow$$

ב.  $\alpha = 2$

$$\left( \begin{array}{ccc|c} 1 & 0 & -(\alpha+3) & 0 \\ 0 & 1 & (\alpha+2) & 1 \\ 0 & 0 & (\alpha-2)(\alpha+3) & (\alpha-2) \end{array} \right) \Rightarrow \begin{array}{l} \text{- אין סוף פתרונות} \\ \left( \begin{array}{ccc|c} 1 & 0 & -5 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 \end{array} \right) \end{array}$$

ג.  $\alpha = -3$

$$\begin{array}{l} \text{- אין פתרון} \\ \left( \begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & -5 \end{array} \right) \end{array}$$

$$\left( \begin{array}{ccc|c} 1 & 0 & -(\alpha+3) & 0 \\ 0 & 1 & (\alpha+2) & 1 \\ 0 & 0 & (\alpha+3) & 1 \end{array} \right) \Rightarrow \left( \begin{array}{ccc|c} 1 & 0 & 0 & \frac{1}{\alpha+3} \\ 0 & 1 & 0 & 1 - \frac{\alpha+2}{\alpha+3} \\ 0 & 0 & 1 & \frac{1}{\alpha+3} \end{array} \right) \Rightarrow \begin{cases} x = \frac{1}{\alpha+3} \\ y = \frac{1}{\alpha+3} \\ z = \frac{1}{\alpha+3} \end{cases}$$

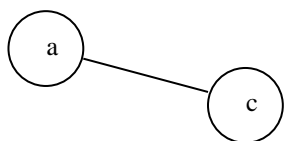
א.  $\alpha \neq 2, \alpha \neq -3$

ב.  $\alpha = 2$

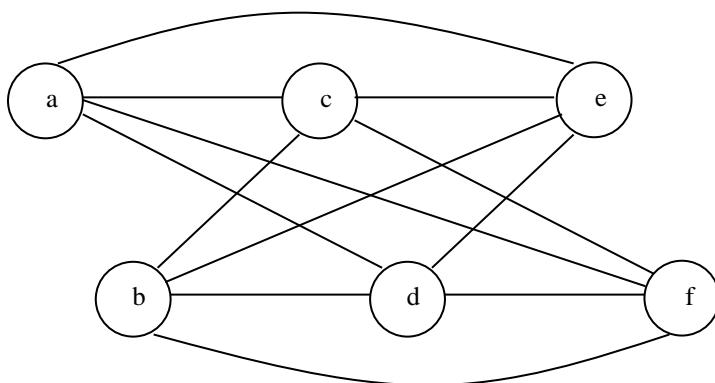
ג.  $\alpha = -3$

**שאלה 10:** הפתרון לשאלה 10 נכתב ע"י לובה סריגין

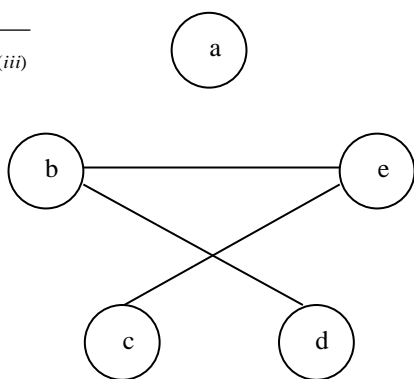
א.  $\overline{G_{(i)}}$



$\overline{G_{(ii)}}$



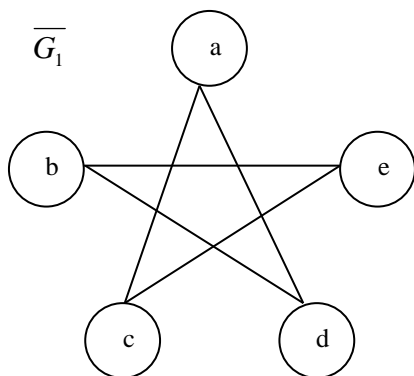
$\overline{G_{(iii)}}$



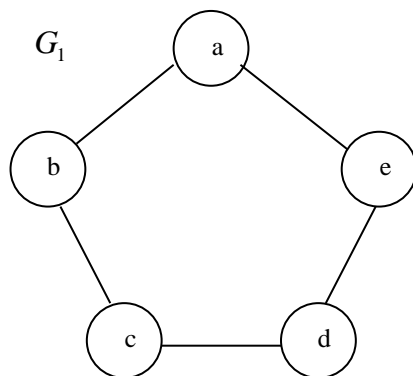
ב. גרף  $\overline{G}$  מכיל את כל הקודקודים שב- $G$ , אין בו קשתות.

ג. מעגל אויילר - מסלול העובר על כל הקשתות בדיוק פעם אחת (במשיכת עפרון אחת).

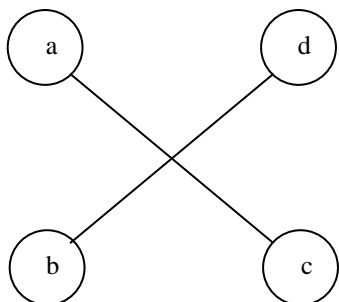
$\overline{G_1}$



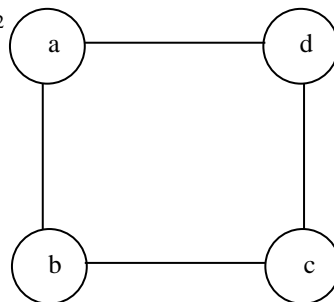
$G_1$



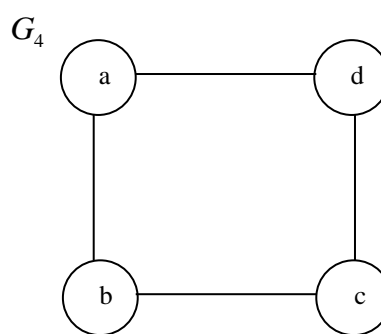
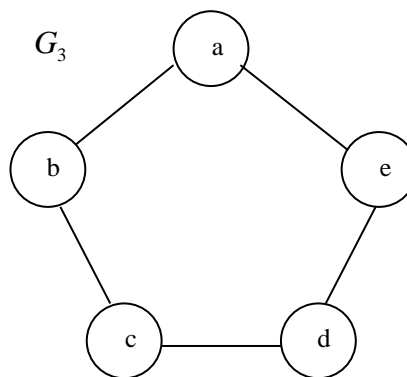
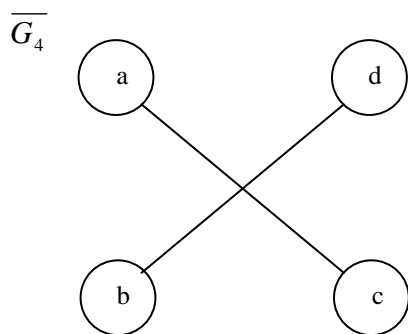
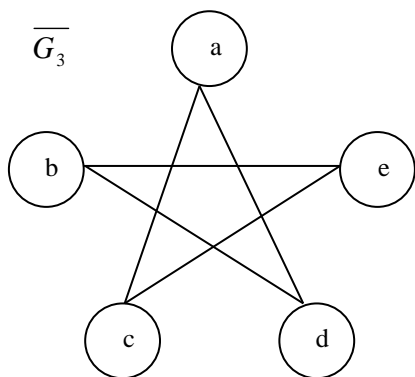
$\overline{G_2}$



$G_2$



ג. מעגל המילטון - מעגל העובר דרך כל הצמתים.



### שאלה 11:

א. (i) השפה הנוצרת ע"י  $G_1$  :  $L = \{ a^{2i} b a^{2j} \mid i, j \geq 0 \}$

(ii) השפה הנוצרת ע"י  $G_1'$  :  $L = \{ a^i b a^j \mid i, j \geq 0 \}$

ב. (i) השפה הנוצרת ע"י  $G_2$  :  $L = \{ a^i b a^j \mid i, j > 0 \}$

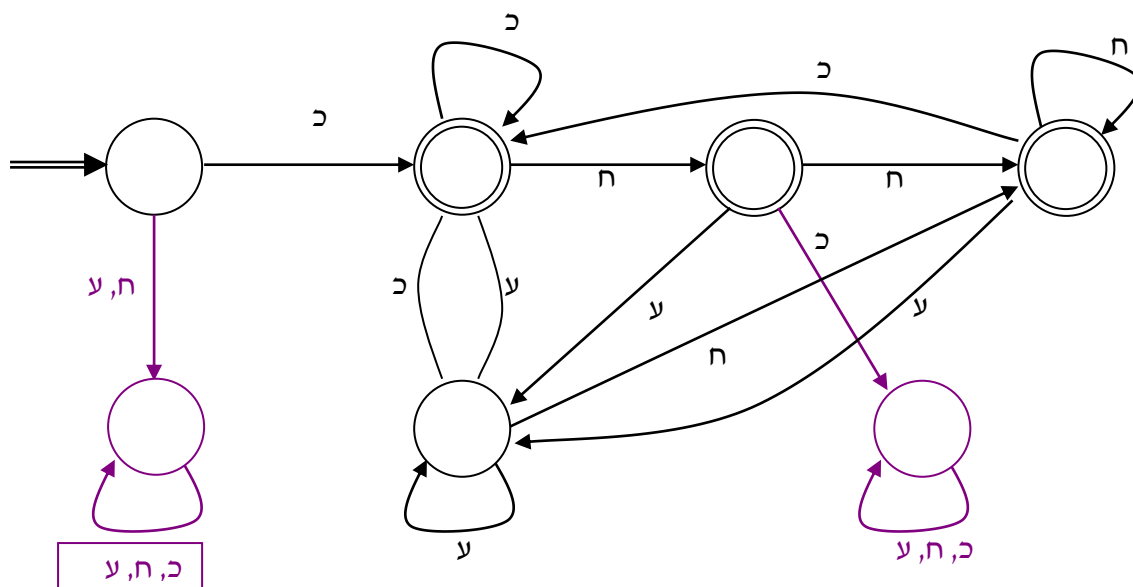
(ii) השפה הנוצרת ע"י  $G_2'$  :  $L = \{ a^i b a^j \mid i, j > 0 \}$

ג. (i) השפה הנוצרת ע"י  $G_3$  :  $L = \{ a^{2i} b a^{2j} \mid i, j \geq 0 \}$

(ii) השפה הנוצרת ע"י  $G_3'$  :  $L = \{ a^i b a^j \mid i, j \geq 0 \}$

**שאלה 12:**

- א. נגדיר: כ = כלב  
ח = חתול  
ע = עכבר
- א"ב השפה:  $\Sigma = \{כ, ח, ע\}$



ב. כללי האריה:

- אין עכבר אחרי חתול
  - אין חתול אחרי עכבר
- ובצורה כללית יותר: אין חתול ליד עכבר.

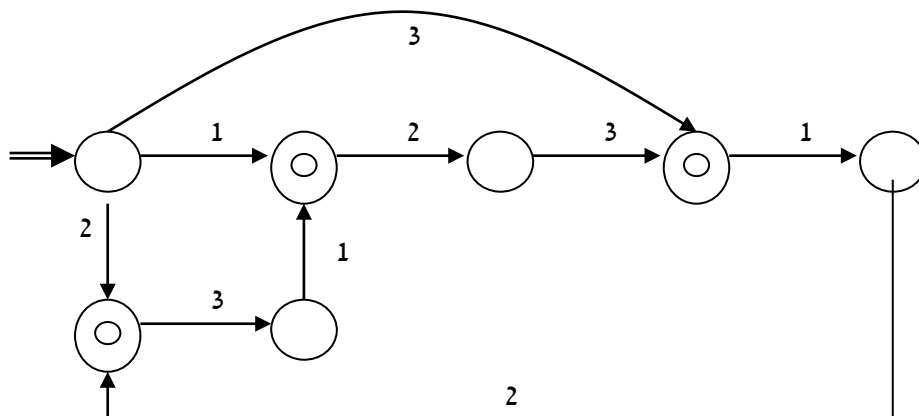
פרק ב'

מודלים חישוביים

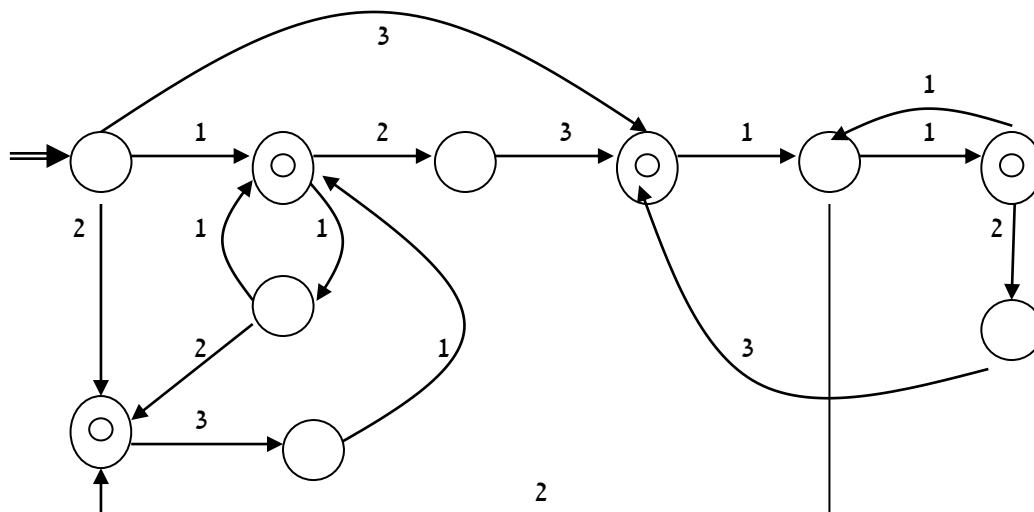
הפתרון לפרק זה נכתב ע"י רחל לוזמר.

תרגיל 13:

א.



ב.



שאלה 14:

- א. (1)  $w = 011100$ , מספר ה-0 שווה למספר ה-1 והוא אינו 5.  
 (2)  $|w| = 8$ ,  $w = 00111111$ , המילה שייכת ל- $L1$ , אורך המילה גדול מ-5, אבל המילה לא שייכת ל- $L2$  כי מספר ה-1 הוא 6 ולא קטן מ-5.  
 (3)  $w = 11110111$ ,  $y = 011$ ,  $x = 11$   
 המילה שייכת ל- $L5$  אבל לא שייכת ל- $L2$  כי מספר ה-1 במילה הוא 7.

- ב. (i)  $\overline{L2} = \{w \mid \#_1(w) \geq 5\}$   
 (ii) כל המילים שבהם מספר ה-0 שונה מ-5 או מספר ה-1 שונה מ-5.  
 $\overline{L3} = \{w \mid \#_0(w) \neq 5\} \cup \{w \mid \#_1(w) \neq 5\}$

- ג. (i)  $L1 \cap L2 \neq \Phi$   
 $L1 \cap L2 = \{w \mid |w| > 5, \#_1(w) < 5\}$   
 למשל  $w = 010101$  היא בחיתוך.

- (ii)  $w = \varepsilon \Rightarrow w \in \overline{L3}$   
 $w \in L4, \#_0(w) = \#_1(w) = 0$   
 $w \notin \overline{L4} \Rightarrow \overline{L3} \not\subset \overline{L4}$

- (iii)  $L4 \cdot L4 = \{w \mid \#_0(w) = \#_1(w)\} \cdot \{\{w_1 \mid \#_0(w_1) = \#_1(w_1)\}\}$   
 $= \{w_2 = w \cdot w_1 \mid \#_0(w_2) = \#_1(w_2)\} = L4$   
 $L4 \cdot L4 \neq L4$  לכן

(iv)

$$L5 \cap L3 \neq \Phi$$

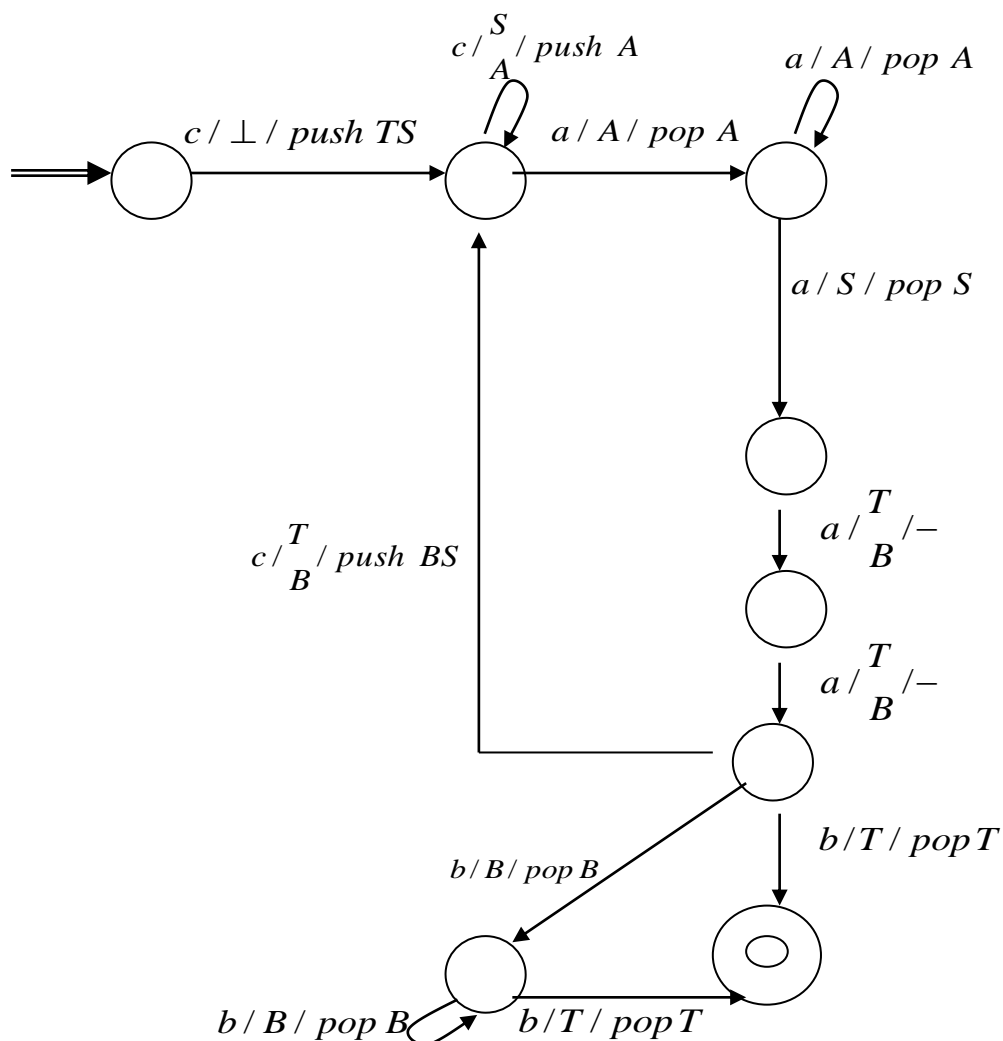
$$L5 \cap L3 = \{w \mid w = xxy, 0 < |x| < 5\} \cap \{w \mid \#_0(w) = 5, \#_1(w) = 5\}$$

$$\{w \mid w = xxy, 0 < |x| < 5, \#_0(w) = 5, \#_1(w) = 5\}$$

לדוגמא מילה בחיתוך:  $x = 01$   
 $y = 000111$   
 $w = 0101000111$

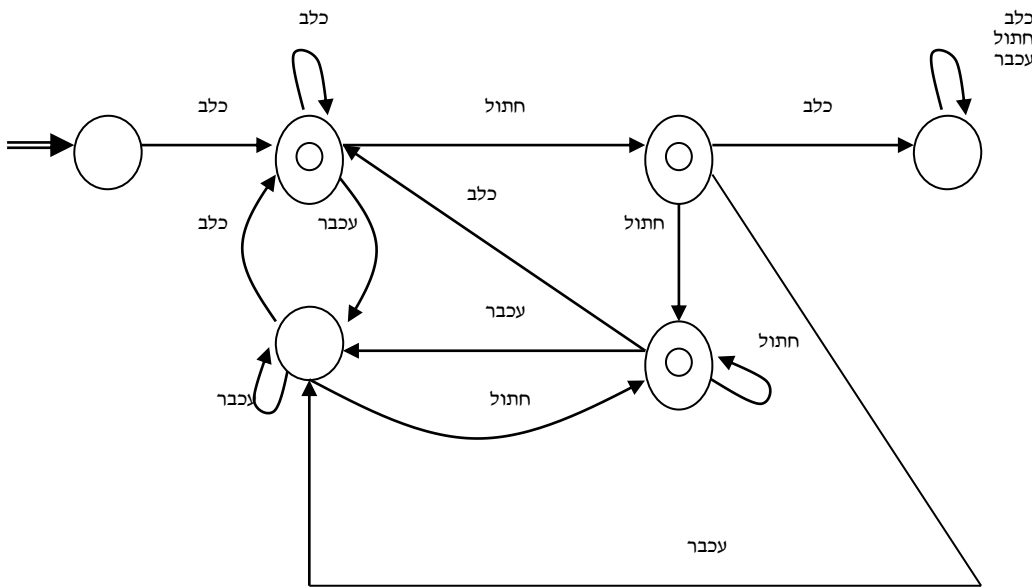


שאלה 15:



**שאלה 16:**

א.



ב. הכללים החדשים הם: חתול לא יכול להיות אחרי עכבר, ועכבר לא יכול להיות אחרי חתול.

## פרק ב'

### תכנות מונחה עצמים Java

הפתרון לפרק זה נכתב ע"י ...

#### תרגיל 17:

א.

```
public class Program
{
    public static void main(String[] args)
    {
        TransportationCompany company1 = new TransportationCompany();

        Vehicle boat = new Boat ("sea ", 50);
        company1.addVehicle (boat);

        company1.addVehicle (new Train (150, 6));

        company1.display();
    }
}
```

אפשר כך (הוספת סירה)  
ואפשר גם כך (הוספת רכבת)

ב. נוסף בכל מחלקה פעולה toString() מתאימה:

במחלקה Vehicle:

```
public String toString()
{
    return this.type + "\t" + this.way +
           "\t max speed : " + this.maxSpeed;
}
```

במחלקה Train:

```
public String toString()
{
    return "Train: \t\t" + super.toString() +
           "\t num of carriages: " + this.numOfCarriages;
}
```

במחלקה Airplane:

```
public String toString()
{
    return "Airplane: \t" + super.toString() +
           "\t max height: " + this.maxHeight;
}
```

במחלקה Boat: (למעשה אפשר לוותר על הפעולה במחלקה זו כי אין לה תכונות נוספות מעבר לאלו שב-Vehicle. הפעולה הוספה למחלקה כדי שגם כלי השיט יציג את שם המחלקה שלו)

```
public String toString()
{
    return "Boat: \t\t" + super.toString();
}
```

ג. הוספת קרונות רכבת הינה באחריותה של הרכבת.  
הוספת קרונות לכל הרכבות שבצי כלי הרכב של החברה הינו באחריותה של החברה. רק במחלקה זו יש גישה ישירה למערך כלי הרכב, ולכן נוסיף את הפעולה במחלקה TransportationCompany :

```
//--- הוספת n שבחברה הרכבות לכל קרונות ---  
public void addCarriages (int n)  
{  
    for (int i = 0 ; i < this.counter ; i++)  
    {  
        if (this.vehicles[i] instanceof Train)  
            ((Train) this.vehicles[i]).incNumOfCariaggess (n);  
    }  
}
```

תרגיל 18:

תרגיל 19:

שאלה 20:

פרק ב'

תכנות מונחה עצמים C#

הפתרון לפרק זה נכתב ע"י ...

תרגיל 21:

תרגיל 22:

תרגיל 23:

שאלה 24: