

מדעי המחשב ה'

פתרון בחינת הבגרות

פרק א - עיצוב תכנה

שאלה 1:

א. טבלת מעקב למשפט הזימון: סוד1(86, 31547)

n1	n2	n1 = 0, n2 = 0	n1 = 0, n2 ≠ 0	n1 ≠ 0, n2 = 0	ערך מוחזר
86	31547	F	F	F	3
8	3154	F	F	F	3
0	315	F	T		1 + 2 → 3
0	31	F	T		1 + 1 → 2
0	3	F	T		1 + 0 → 1
0	0	T			0

- ב. הערך המוחזר הוא 3. האלגוריתם מחזיר את ההפרש במספר הספרות של n1 ו-n2
- ג. טבלת מעקב למערך a ולזימון: סוד2(a, 5, 1)

	1	2	3	4	5
a	9321	345	296	7	98

n	k	k = n - 1 (k = 4)	סוד1(a[k], a[k+1])	ערך מוחזר max(sod1, sod2)
5	1	F	סוד1(9321, 345) → 1	max(1, 2) → 2
5	2	F	סוד1(345, 296) → 0	max(0, 2) → 2
5	3	F	סוד1(296, 7) → 2	max(2, 1) → 2
5	4	T	סוד1(7, 98) → 1	1

הערך המוחזר הוא 2.

- ד. סוד2 מחזיר את הפרש מספר-הספרות הגדול ביותר בין שני תאים סמוכים במערך החל ממקום k ועד המקום N (סוף מהערך).

שאלה 2:

מסעדת "טעמים 22"

א. ייצוג בשפת פסקל:

type

<pre>ingredientType = string;</pre>	<p>מרכיב-מזון = מחרוזת</p>
<pre>list_info_type = ingredientType; (שורה זו תרשם ביחידת ספריה List) או: ingrediantList = array [1..15] of ingredientType</pre>	<p>רשימת-המרכיבים-למנה = רשימה שכל איבר בה מטיפוס מרכיב-מזון. או: מערך בגודל 15 של מרכיבי-מזון.</p>
<pre>course = record code : integer; dishName : string; L : list_type; end; או: course = record code : integer; dishName : string; L : ingrediantList; end;</pre>	<p>מנה-במסעדה = רשומה קוד-מנה - מספר שלם שם-המנה - מחרוזת סוגי-מזון = רשימת המרכיבים למנה</p>
<pre>menu = record menuName : string; arr : array [1..22] of course ; end;</pre>	<p>תפריט-במסעדה = רשומה שם-התפריט - מחרוזת מערך בגודל 22 של מנות-במסעדה</p>

א. ייצוג בשפת C:

<pre>typedef char * ingredientType ;</pre>	<p>מרכיב-מזון = מחרוזת</p>
<pre>typedef ingredientType list_info_type; (שורה זו תרשם ביחידת ספריה List) או: ingredientType ingrediantList [15];</pre>	<p>רשימת-המרכיבים-למנה = רשימה שכל איבר בה מטיפוס מרכיב-מזון. או: מערך בגודל 15 של מרכיבי-מזון.</p>
<pre>typedef struct { int code; char * dishName; list_type L; } course ; או: typedef struct { int code; char * dishName; ingrediantList L ; } course;</pre>	<p>מנה-במסעדה = רשומה קוד-מנה - מספר שלם שם-המנה - מחרוזת סוגי-מזון = רשימת המרכיבים למנה</p>
<pre>typedef struct { char * menuName; course arr[22]; } menu ;</pre>	<p>תפריט-במסעדה = רשומה שם-התפריט - מחרוזת מערך בגודל 22 של מנות-במסעדה</p>

ב.

מרכיב-F-לא-במנה (M, f)

{ הפעולה מקבלת תפריט M ומרכיב-מזון f ומדפיסה את כל המנות ש-f אינו מרכיב בהן (את כל המנות שניתן להכין).
 הנחה: התפריט מאותחל ותקין. }

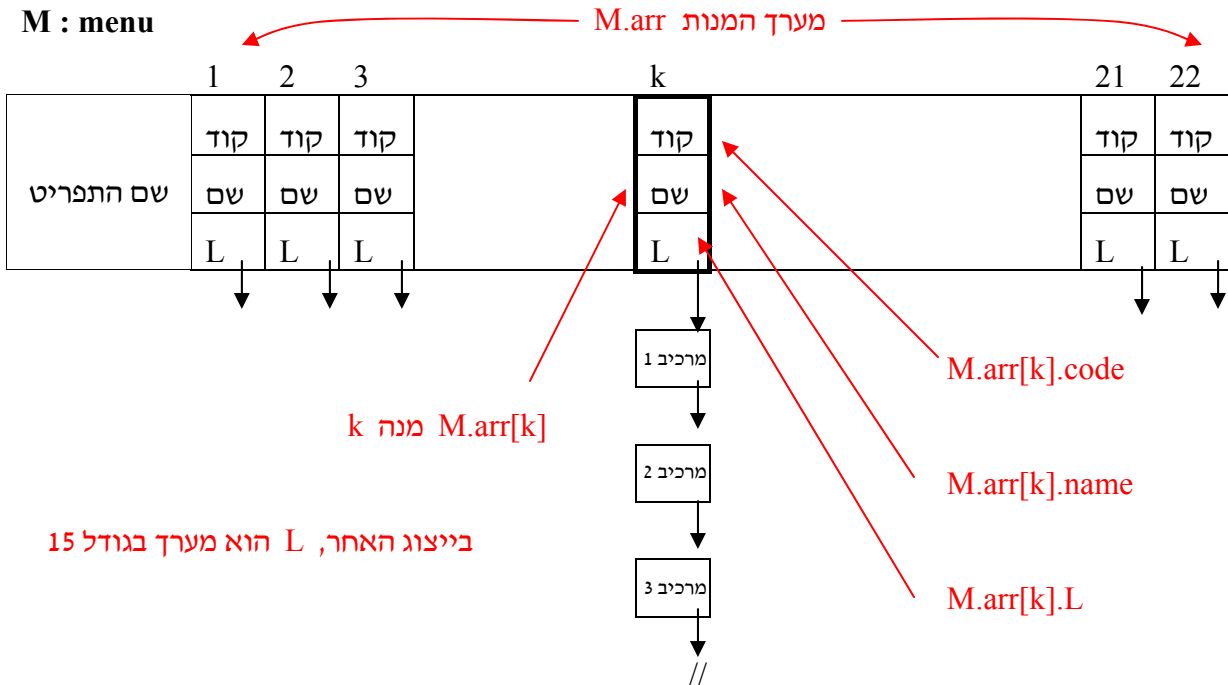
(1) עבור i מ-1 ועד 22 בצע:

{ L מקבל את רשימת המרכיבים למנה i } $L \leftarrow M.arr[i].L$ (1.1)

(1.2) אם לא נמצא-ברשימה (L, f) אזי

הדפס את $M.arr[i].code$

M : menu



נמצא-ברשימה (L, f)

{ פעולה המחזירה 'אמת' אם f הוא איבר ברשימת מרכיבי המזון L ו-'שקר' אחרת.
 הנחה: L מאותחל. }

(1) עוקב-ברשימה (עוגן-רשימה (L, L) $p \leftarrow (L, L)$

(2) כל עוד $p \neq$ סוף-רשימה (L) בצע:

(2.1) אחזר-מרשימה (L, p) $x \leftarrow (L, p)$

(2.2) אם $x = f$ אזי

החזר 'אמת'

(2.3) עוקב-ברשימה (L, p) $p \leftarrow (L, p)$

(3) החזר 'שקר'

ואם L הוא מערך מרכיבי המזון:

```

    נמצא-ברשימה (L, f)
    {
        פעולה המחזירה 'אמת' אם f הוא איבר במערך מרכיבי המזון L ו-'שקר' אחרת.
        הנחה: L מאותחל.
        i ← 1 (1)
        כל עוד i ≤ 15 וגם L[i] אינו תא ריק, בצע: (2)
            אם L[i] = f אזי החזר 'אמת' (2.1)
            i ← i + 1 (2.2)
        החזר 'שקר' (3)
    }
    
```

פסקל:

```

    {
        הפעולה מקבלת תפריט M ומרכיב-מזון f ומדפיסה את כל המנות ש-f אינו מרכיב בהן (את כל המנות שניתן להכין).
        הנחה: התפריט מאותחל ותקין.
    }
    
```

procedure **Ingredient_F_not_in_course** (M : menu ; f : ingredientType);

```

var   i : integer;
      L : list_type;
begin
    for i := 1 to 22 do
        begin
            L := M.arr[i].L;
            if not found_in_list (L, f) then
                writeln (M.arr[i].code);
        end;
    end;
end;
    
```

```

    {
        פעולה המחזירה 'אמת' אם f הוא איבר ברשימת מרכיבי המזון L ו-'שקר' אחרת.
        הנחה: L מאותחל.
    }
    
```

function **found_in_list** (L : list_type ; f : ingredientType) : boolean;

```

var   p : pos_type;
      x : list_info_type;
      found : boolean;
begin
    found := false;
    p := list_next (L, list_anchor (L));
    while (p <> list_end (L)) and (not found) do
        begin
            list_retrieve (L, p, x);
            if (x = f) then
                found := true;
            else
                p := list_next (L, p);
        end;
    end;
end;
    
```

```

        end;
    found_in_list := found;
end;

```

{ פעולה המחזירה 'אמת' אם f הוא איבר במערך מרכיבי המזון L ו-'שקר' אחרת.
 הנחה : L מאותחל. }

```

function found_in_list (L : ingrediantList ; f : ingredientType) : boolean;
var    i : integer;
        found : boolean;
begin
    found := false;
    i := 1;
    while (i <=15) and (L[i] <> " ) and (not found) do
        if (L[i] = f) then
            found := true
        else
            i := i + 1;
        end if;
    end while;
    found_in_list := found;
end;

```

C:

// הפעולה מקבלת תפריט M ומרכיב-מזון f ומדפיסה את כל המנות ש-f אינו מרכיב בהן
 // (את כל המנות שניתן להכין).
 // הנחה : התפריט מאותחל ותקין.

```

void Ingredient_F_not_in_course (menu M , ingredientType f)
{
    int i;
    list_type L;
    for (i = 0 ; i < 22 ; i++)
    {
        L = M.arr[i].L;
        if (found_in_list (L, f) == 0)
            printf ("%d \n", (M.arr[i].code));
    }
}

```

```
// פעולה המחזירה 'אמת' אם f הוא איבר ברשימת מרכיבי המזון L ו-'שקר' אחרת.  
// הנחה: L מאותחל.
```

```
int found_in_list (list_type L , ingredientType f)  
{  
    pos_type p;  
    list_info_type x;  
    p = list_next (L, list_anchor (L));  
    while (p != list_end (L)  
    {  
        list_retrieve (L, p, x);  
        if (strcmp(x, f) == 0)  
            return 1;  
        p = list_next (L, p);  
    }  
    return 0;  
}
```

```
// פעולה המחזירה 'אמת' אם f הוא איבר במערך מרכיבי המזון L ו-'שקר' אחרת.  
// הנחה: L מאותחל.
```

```
int found_in_list (ingrediantList L , ingredientType f)  
{  
    int i = 0;  
    while (i < 15 && strcmp (L[i] , "") != 0)  
    {  
        if (strcmp (L[i] , f) == 0)  
            return 1;  
        i ++;  
    }  
    return 0;  
}
```

הערה: בפתרון שאלות 2 ו-4 נעשה שימוש בתבניות אלגוריתמיות המופיעות בתבניות בעיצוב תכנה של אוניברסיטת תל-אביב.

מסעדת "טעמים 22": א. ייצוג בשפת Java: (הטבלה שלהלן מתאימה גם ל-C#)

<pre>public class Ingredient { private String ingredient ; }</pre>	<p>מרכיב-מזון = מחרוזת</p>
<pre>public class Course { private int code; private String title; List <Ingredient> ingeds; public static int N = 15; Ingredient [] ingreds; private int last; public Course (int code,String title){ this.code = code; this.title = title; ingreds = new List <Ingredient>(); } or: ingreds = new Ingredient[N]; this.last = 0; } public void addIngredient (String name){ this.ingreds.insert (neme); } or: if (this.last < 15) ingreds[this.last ++] = name; } }</pre> <p style="text-align: right;">או:</p>	<p>מנה במסעדה = קוד מנה = מספר שלם שם המנה = מחרוזת</p> <p>רשימת-המרכיבים-למנה = רשימה שכל איבר בה מטיפוס מרכיב-מזון.</p> <p>או: מערך בגודל 15 של מרכיבי-מזון.</p> <p>הערות: <ul style="list-style-type: none"> • במקום המחלקה Ingredient ניתן להשתמש ב-String. בכל מקום שבו כתוב Ingredient נחליף את הכיתוב ב-String • הקוד שבצבע ירוק אינו חלק שנדרש בפתרון הבחינה, והוא מובא כאן לצרכי הבהרת הפתרון בלבד. </p>
<pre>public class Menu { public static int N = 22; private int last; Course [] arr; public Menu () { this.arr = new Course [N]; this.last = 0; } public void addCourse (Course cor) { if (this.last < N) arr[this.last ++] = cor; } }</pre>	<p>תפריט-במסעדה = שם-התפריט - מחרוזת מערך בגודל 22 של מנות-במסעדה</p>

תודה לעפרה ברנדס מצוות הפיתוח בירושלים.

פתרון בשפת Java:

```
import unit4.collectionsLib.List;
import unit4.collectionsLib.Node;

/**
 * פתרון לשאלה 2 בבגרות קיץ תשס"ז
 * צוות הפיתוח, האוניברסיטה העברית ירושלים @author
 * @version 15/5/2007
 */

public class Course{
    private int code;
    private String name;
    private List<String> ingredients;
    public static final int MAX_INGREDIENTS = 15;

    public boolean containsIngredient(String f)
    {
        Node<String> pos = this.ingredients.getFirst();
        while (pos != null) {
            if (pos.getInfo().equals(f))
                return true;
            pos = pos.getNext();
        }
        return false;
    }

    public int getCode()
    {
        return this.code;
    }
}

/**
 * פתרון לשאלה 2 בבגרות קיץ תשס"ז
 * צוות הפיתוח, האוניברסיטה העברית ירושלים @author
 * @version 15/5/2007
 */

public class Menu{
    private String name;
    private Course[] courses = new Course[22];

    public void printAvailableCourses(String f)
    {
        for (int i=0; i<22; i++)
        {
            if ((courses[i]!= null) && (!courses[i].containsIngredient(f)))
                System.out.println(courses[i].getCode());
        }
    }
}
```

```
    }  
  }  
}
```

תודה לעפרה ברנדס מצוות הפיתוח בירושלים.

פתרון בשפת C#:

```
using System;  
using System.Collections.Generic;  
using System.Text;  
  
namespace Bagrut  
{  
    /// <summary>  
    /// פתרון לשאלה 2 בבגרות קיץ תשס"ז  
    /// </summary>  
    /// <author> צוות הפיתוח, האוניברסיטה העברית ירושלים </author>  
    /// <version> 17/5/2007 </version>  
    public class Course  
    {  
        private int code;  
        private string name;  
        private List<string> ingredients;  
        public const int MAX_INGREDIENTS = 15;  
  
        public bool ContainsIngredient(string f)  
        {  
            Node<string> pos = this.ingredients.GetFirst();  
            while (pos != null)  
            {  
                if (pos.GetInfo().Equals(f))  
                    return true;  
                pos = pos.GetNext();  
            }  
            return false;  
        }  
  
        public int GetCode()  
        {  
            return this.code;  
        }  
    }  
}
```

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Bagrut
{
    /// <summary>
    /// פתרון לשאלה 2 בבגרות קיץ תשס"ז
    /// </summary>
    /// <author> צוות הפיתוח, האוניברסיטה העברית ירושלים </author>
    /// <version> 17/5/2007 </version>
    public class Menu
    {
        private string name;
        private Course[] courses = new Course[22];

        public void PrintAvailableCourses(string f)
        {
            for (int i = 0; i < 22; i++)
            {
                if ((courses[i] != null) &&
                    (!courses[i].ContainsIngredient(f)))
                    Console.WriteLine(courses[i].GetCode());
            }
        }
    }
}
```

שאלה 3:

מסלול-אחיד? (T)

{ פעולה המחזירה 'אמת' אם קיים מסלול, המתחיל בשורש העץ T ומסתיים באחד העלים שלו, וכל ערכי הצמתים בו זהים, ו-'שקר' אחרת. הנחה: העץ T מאותחל. }

(1) החזר: **קיים-מסלול-אחיד? (T, 1)** א **קיים-מסלול-אחיד? (T, 2)**

קיים-מסלול-אחיד? (T, x)

{ פעולה המחזירה 'אמת' אם קיים מסלול, המתחיל בשורש העץ T ומסתיים באחד העלים שלו, וערכי כל הצמתים בו הם x, ו-'שקר' אחרת. הנחה: העץ T מאותחל. }

- (1) אם עץ-ריק? (T) החזר 'שקר'
- (2) אם אחזר-שורש (T) $x \neq$ החזר 'שקר'
- (3) אם עלה? (T) החזר 'אמת' { בשלב זה כבר ברור שערך הצומת הוא x }
- (4) החזר **קיים-מסלול-אחיד? (x, תע"ש (T))** א **קיים-מסלול-אחיד? (x, תע"י (T))**

עלה? (T)

- { פעולה המחזירה 'אמת' אם T עלה, ו-'שקר' אחרת. הנחה: T מאותחל }
- (1) אם עץ-ריק? (T) החזר 'שקר'
- (2) אם עץ-ריק? (תע"ש (T)) וגם עץ-ריק? (תע"י (T)) החזר 'אמת'
- (3) החזר 'שקר'.

פתרון 2: נכתב אז יבי איתן ראט:

מסלול-אחיד? (T)

{ הפעולה מחזירה 'אמת' אם קיים מסלול אחיד (לפי הגדרת השאלה) בעץ הבינארי T, ו-'שקר' אחרת. הנחה: העץ מאותחל. }

- (1) אם (עץ-ריק? (T)) או (עלה? (T)) אזי (1.1) החזר: 'אמת'
- (2) (לא עץ-ריק? (תע"ש (T))) וגם (אחזר-שורש (T)) = (אחזר-שורש (תע"ש (T))) וגם (מסלול-אחיד? (תע"ש (T))) \leftarrow Left
- (3) (לא עץ-ריק? (תע"י (T))) וגם (אחזר-שורש (T)) = (אחזר-שורש (תע"י (T))) וגם (מסלול-אחיד? (תע"י (T))) \leftarrow Right
- (4) החזר: (Left) או (Right)

בתרון 3: נכתב על ידי ראוי גבאי:

מסלול-אחיד? (T)

- פעולה מקבלת עץ בינארי ומחזירה 'אמת' אם קיים מסלול אחיד אחרת יוחזר 'שקר'.
 הנחה: T עץ בינארי מאותחל.
- (1) אם עץ-ריק? (T) אזי החזר 'שקר'
 (2) אם עלה? (T) אזי החזר 'אמת'
 (3) אחזר-שורש (T) ← x
 (4) אם לא עץ-ריק? (תת-עץ-שמאלי (T)) וגם לא עץ-ריק? (תת-עץ-ימני(T)) וגם אחזר-שורש (תת-עץ-שמאלי(T)) x ≠ וגם אחזר-שורש (תת-עץ-ימני(T)) x ≠ אזי
- (4.1) החזר 'שקר'
 (5) אם לא עץ-ריק? (תת-עץ-שמאלי(T)) וגם (עץ-ריק? (תת-עץ-ימני(T)) וגם אחזר-שורש (תת-עץ-שמאלי(T)) x ≠ אזי
- (5.1) החזר 'שקר'
 (6) אם (עץ-ריק(תת-עץ-שמאלי(T)) וגם (לא עץ-ריק(תת-עץ-ימני(T))) וגם אחזר-שורש (תת-עץ-ימני(T)) x ≠ אזי
- (6.1) החזר 'שקר'
 (7) החזר: מסלול-אחיד? (תת-עץ-שמאלי(T)) או מסלול-אחיד? (תת-עץ-ימני(T))

בתרון 4: נכתב על ידי גברה כן-אורי:

אמנם תור ירד במיקוד, אבל הנה עוד רעיון לפתרון - אלגוריתם מסלול-אחיד, לא רקורסיבי, המשתמש בתור:

מסלול-אחיד (T)

- פעולה מקבלת עץ בינארי ומחזירה 'אמת' אם קיים מסלול אחיד אחרת יוחזר 'שקר'.
 הנחה: T עץ בינארי מאותחל.
- (1) אם לא עץ-ריק? (T) אזי:
- (1.1) אחזר-שורש (T) ← x
 (1.2) אתחל תור ← Q
 (1.3) הכנס-לתור (Q,T)
 (1.4) כל עוד לא תור-ריק? (Q), בצע:
- (1.4.1) הוצא-מתור(Q) ← T1
 (1.4.2) אם עלה? (T1), החזר 'אמת'
 (1.4.3) אם לא עץ-ריק? (תע"ש(T1)), אזי:
- (1.4.3.1) אם אחזר-שורש(תע"ש(T1)) = x אזי
 הכנס-לתור (תע"ש(T1),Q)
 (1.4.4) אם לא עץ-ריק? (תע"י(T1)), אזי:
- (1.4.4.1) אם אחזר-שורש(תע"י(T1)) = x אזי
 הכנס-לתור (תע"י(T1),Q)
- (2) החזר 'שקר'

שאלה 4:

טלפון נייד:

א. ייצוג:

מספר-טלפון = מחרוזת.
 יומן-שיחות-נכנסות = רשומה
 L - רשימת מספרי הטלפון שביומן השיחות הנכנסות: רשימה שכל איבר בה מטיפוס מספר-טלפון.
 maxCalls - מספר שלם ולא שלילי המייצג את המספר המקסימלי של מספרי-טלפון שניתן לשמור ביומן השיחות הנכנסות.
 currentCalls - מספר שלם ולא שלילי המייצג את מספר מספרי הטלפון השמורים בזמן מסויים ביומן השיחות הנכנסות.

ייצוג בשפת פסקל:

```
type
    phoneNum = string [10];
    list_info_type = phoneNum ;

    incomingCallLog = record
        L : list_type;
        maxCalls , currentCalls : integer;
    end;
```

ביחידת ספרייה רשימה נרשום:
 בתכנית נגדיר:

ייצוג בשפת C:

```
char phoneNum [10];
typedef phoneNum list_info_type ;

typedef struct
{
    list_type L;
    int maxCalls , currentCalls;
} incomingCallLog ;
```

ביחידת ספרייה רשימה נרשום:
 בתכנית נגדיר:

משתנים:
 d = יומן השיחות הנכנסות
 tel = מספר טלפון.

ב. תת תכנית לעדכון יומן השיחות הנכנסות:

תהליך עדכון שיחה נכנסת:

האם tel נמצא ברשימת השיחות הנכנסות?

אם כן - הוצאתו מהמיקום הנוכחי והעברתו לתחילת הרשימה.
אם לא -

האם יש מקום פנוי ביומן?

אם כן - הכנסת tel לתחילת הרשימה ועדכון currentCalls

אם לא - מחיקת השיחה האחרונה ברשימה והכנסת tel לתחילת הרשימה.

פעולות עזר:

אתר-ברשימה (L, tel) המחזיר את מקומו של tel ברשימה L. אם לא קיים tel יחזיר את סוף-רשימה.

הכנס-להתחלה (L, tel) המכניס את מספר הטלפון לתחילת הרשימה.

מחק-מהסוף (L) המוחק את מספר הטלפון האחרון ברשימה.

יש-מקום-פנוי? (d) המחזיר 'אמת' אם יש מקום פנוי ביומן השיחות, ו-'שקר' אחרת.

שיחה-נכנסת (d, tel)

} ט.כניסה: יומן שיחות נכנסות d ומספר טלפון tel.

ט.יציאה: יומן השיחות מעודכן.

{ הנחה: d מאותחל ותקין.

(1) אתר-ברשימה (d.L, tel) $p \leftarrow$

(2) אם $p \neq$ סוף-רשימה (d.L) אזי -

(2.1) הוצא-מרשימה (d.L, p)

(2.2) הכנס-להתחלה (d.L, tel)

(3) אחרת - { tel אינו ברשימה }

(3.1) אם יש-מקום-פנוי? (d) אז -

(3.1.1) הכנס-להתחלה (d.L, tel)

(3.1.2) $d.currentCalls \leftarrow d.currentCalls + 1$

(3.2) אחרת -

(3.2.1) מחק-מהסוף (d.L)

(3.2.2) הכנס-להתחלה (d.L, tel)

פסקל:

```

{
    ט.כניסה : יומן שיחות נכנסות d ומספר טלפון tel.
    ט.יציאה : יומן השיחות מעודכן.
    הנחה : d מאותחל ותקין.
}

procedure incommingCall (var d : incoingCallLog; tel : phoneNum)
var p : pos_type;
begin
    p := list_find_place (d.L, tel);
    if (p <> list_end (d.L)) then
        begin
            list_delete (d.L, p);
            list_insert_top (d.L, tel);
        end
    else
        begin
            if (avail_place (d)) then
                begin
                    list_insert_top (d.L, tel);
                    d.currentCalls := d.currentCalls + 1;
                end
            else
                begin
                    list_delete_bottom (d.L);
                    list_insert_top (d.L, tel);
                end;
            end;
        end;
end;

{ פעולה המחזירה את מקומו של tel ברשימה מאותחלת L, אם לא נמצא יוחזר המקום סוף-רשימה. }
function list_find_place (L : list_type; tel : list_info_type) : pos_type;
var p : pos_type;
    x : list_info_type;
    found : boolean;
begin
    found := false;
    p := list_next (L, list_anchor (L));
    while (p <> list_end (L)) and not found do
        begin
            list_retrieve (L, p, x);
            if (x = tel) then
                found := true
            else
                p := list_next (L, p);
            end;
        end;
end;

```



```
list_find_place := p;
end;
```

```
{ פעולה המכניסה את tel לתחילתה של רשימה מאותחלת L }
procedure list_insert_top (var L : list_type , tel : list_info_type);
begin
    list_insert (L, list_anchor (L), tel);
end;
```

```
{ פעולה המוחקת את האיבר שנמצא בסוף הרשימה המאותחלת L }
procedure list_delete_bottom (var.L : list_type);
var    p : pos_type;
begin
    p := list_prev (L, list_end(L));
    list_delete (L, p);
end;
```

```
{ פעולה המחזירה 'אמת' אם יש מקום פנוי ביומן השיחות המאותחל d , ו-'שקר' אחרת. }
function avail_place (d : incomingCallLog) : boolean;
begin
    avail_place := (d.currentCalls < d.maxCalls);
end;
```

C:

```

#define TRUE 1
#define FALSE 0

// ט.כניסה: יומן שיחות נכנסות d ומספר טלפון tel.
// ט.יציאה: יומן השיחות מעודכן.
// הנחה: d מאותחל ותקין.

void incomingCall (incoingCallLog * d, phoneNum tel)
{
    pos_type p;
    p = list_find_place (d.L, tel);
    if (p != list_end (d.L))
    {
        list_delete (&d.L, &p);
        list_insert_top (&d.L, tel);
    }
    else
    {
        if (avail_place (d))
        {
            list_insert_top (&d.L, tel);
            d.currentCalls := d.currentCalls + 1;
        }
        else
        {
            list_delete_bottom (&d.L);
            list_insert_top (&d.L, tel);
        }
    }
}

// פעולה המחזירה את מקומו של tel ברשימה מאותחלת L, אם לא נמצא יוחזר המקום סוף-רשימה.
pos_type list_find_place (list_type L, list_info_type tel)
{
    pos_type p;
    list_info_type x;
    p = list_next (L, list_anchor (L));
    while (p != list_end (L))
    {
        list_retrieve (L, p, &x);
        if (strcmp(x, tel) == 0)
            return p;
        p = list_next (L, p);
    }
}

```

```
    }
    return p;
}

// L המכניסה את tel לתחילתה של רשימה מאותחלת L
void list_insert_top (list_type *L , list_info_type tel)
{
    list_insert (*L, list_anchor (*L), tel);
}

// L המוחקת את האיבר שנמצא בסוף הרשימה המאותחלת L
void list_delete_bottom (list_type *L)
{
    pos_type p;
    p = list_prev (*L, list_end(*L));
    list_delete (L, p);
}

// פעולה המחזירה 'אמת' אם יש מקום פנוי ביומן השיחות המאותחל d , ו-'שקר' אחרת.
int avail_place (incomingCallLog d)
{
    if (d.currentCalls < d.maxCalls)
        return TRUE;
    return FALSE;
}
```

פתרון בשפת Java:

תודה לעפרה ברנדס מצוות הפיתוח בירושלים.

```
import unit4.collectionsLib.List;
import unit4.collectionsLib.Node;

/**
 * פתרון לשאלה 4 בבגרות קיץ תשס"ז
 * צוות הפיתוח, האוניברסיטה העברית ירושלים @author
 * @version 15/5/2007
 */
public class IncomingCallsDiary
{
    private List<String> incomingCalls;
    private int maxCalls;
    private int currentCalls;
    public void addIncomingCall(String tel)
    {
        boolean telWasAlreadyInDiary = false;
        Node<String> prev = null;
        Node<String> curr = this.incomingCalls.getFirst();
        while (curr != null)
        {
            if (curr.getInfo().equals(tel))
            {
                this.incomingCalls.remove(curr);
                telWasAlreadyInDiary = true;
            }
            prev = curr;
            curr = curr.getNext();
        }
        if (!telWasAlreadyInDiary)
        {
            if (this.currentCalls < this.maxCalls)
                this.currentCalls++;
            else
                this.incomingCalls.remove(prev);
        }
        this.incomingCalls.insert(null,tel);
    }
}
```

תודה לעפרה ברנדס מצוות הפיתוח בירושלים.

פתרון בשפת C#:

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Bagrut
{
    /// <summary>
    /// פתרון לשאלה 4 בבגרות קיץ תשס"ז
    /// </summary>
    /// <author> צוות הפיתוח, האוניברסיטה העברית ירושלים </author>
    /// <version> 17/5/2007 </version>
    public class IncomingCallsDiary
    {
        private List<string> incomingCalls;
        private int maxCalls;
        private int currentCalls;

        public void AddIncomingCall(string tel)
        {
            bool telWasAlreadyInDiary = false;
            Node<string> prev = null;
            Node<string> curr = this.incomingCalls.GetFirst();
            while (curr != null)
            {
                if (curr.GetInfo().Equals(tel))
                {
                    this.incomingCalls.Remove(curr);
                    telWasAlreadyInDiary = true;
                }
                prev = curr;
                curr = curr.GetNext();
            }
            if (!telWasAlreadyInDiary)
            {
                if (this.currentCalls < this.maxCalls)
                    this.currentCalls++;
                else
                    this.incomingCalls.Remove(prev);
            }
            this.incomingCalls.Insert(null, tel);
        }
    }
}
```

פרק ב'

מערכות מחשב ואסמבלר

הפתרון לפרק זה נכתב ע"י עידן פרייברג (תלמיד מגמת הנדסת תכנה, מקיף חי ראשלי"צ)

תערה: הפתרון מובא כלשונו, וללא הגהה.

תרגיל 5:

תרגיל 6:

תרגיל 7:

סעיף א'

- i. הקטע אינו מבצע את הנדרש. בהוראת החיסור (SUB BX,M) האופרנדים מסוגים שונים (לא מתאימים) (BX – מסוג מילה, N=N[0] שהוא מסוג בית) דבר שאינו יתכן.
- ii. הקטע אינו מבצע את הנדרש. ההוראה LEA AX,[M] לא תקינה.
- iii. הקטע מבצע את הנדרש.
- iv. הקטע אינו מבצע את הנדרש. נתון כי אינדקס המערך מקבל ערכים בין 0 לבין 49 ולכן אין צורך להגדיל את BX (INC BX)

סעיף ב'

- i. הקטע אינו מבצע את המשימה. לאחר ההוראה DEC AX חסרה הוראת הקפיצה JMP A2.
- ii. הקטע אינו מבצע את המשימה. השגיאה היא בהוראה JGE A1 באלגוריתם נתון כי רק אם $X > 50$ עלינו להגדיל את ערכו ב-1. לכן צריך להיות JG A1.
- iii. הקטע מבצע את המשימה.
- iv. הקטע אינו מבצע את המשימה. יש להחליף בין הוראת DEC AX לבין INC BX. (הקטע הנ"ל מבצע את ההפך)

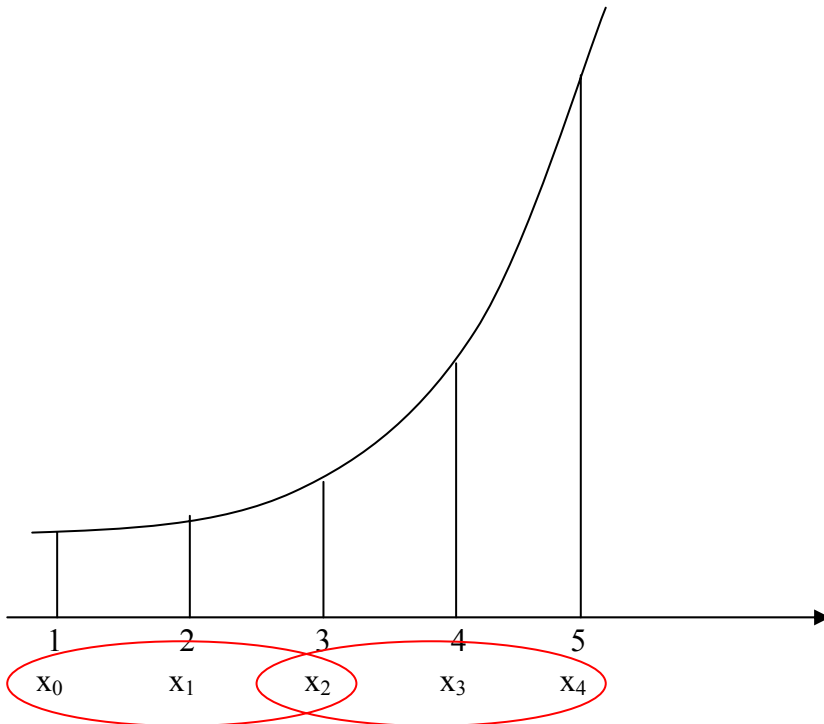
שאלה 8:

```
.MODEL SMALL
.STACK 100H
    N EQU 11 ; גודל המערך N
.DATA
    A DB N dup(?)
.CODE
START: MOV AX,@DATA
        MOV DS,AX
        LEA BX,A    ; BX ← A הכתובת של מערך
        MOV SI,0
        MOV AL,0    ; AL יכיל את סכום האיברים הסמוכים המקסימלי
        MOV CX,N-1
AGAIN: MOV AH,[BX+SI]
        INC SI
        MOV DH,[BX+SI]
        ADD AH,DH
        CMP AH,AL
        JLE CONT
        XCHG AL,AH
CONT:   LOOP AGAIN

MOV AH,4CH
INT 21H
END START
```

פרק ב'
תורת המחשב

שאלה 9:



נתון: $f(x) = x^4$
בקטע: $[1,5]$

$a = 1$
 $b = 5$
 $n = 4$

א. חישוב השטח תחת גרף הפונקציה:

$$S_{[i,i+2]} = \frac{(b-a)}{3n} (f(x_i) + 4f(x_{i+1}) + f(x_{i+2}))$$

$$S_{[0,2]} = \frac{(b-a)}{3n} (f(x_0) + 4f(x_1) + f(x_2)) = \frac{4}{12} (1^4 + 4 \cdot 2^4 + 3^4) = 48 \frac{2}{3}$$

$$S_{[2,4]} = \frac{(b-a)}{3n} (f(x_2) + 4f(x_3) + f(x_4)) = \frac{4}{12} (3^4 + 4 \cdot 4^4 + 5^4) = 576 \frac{2}{3}$$

$$S_{[0,2]} + S_{[2,4]} = 48 \frac{2}{3} + 576 \frac{2}{3} = 625 \frac{1}{3}$$

הערה: לצורך ההשוואה, חישוב האינטגרל מתחת לגרף הפונקציה בתחום הנתון הוא: 624

ב. אלגוריתם לחישוב השטח לפי השיטה המתוארת:

טבלת משתנים:

שם משתנה	טיפוס המשתנה	תפקיד המשתנה
a, b	מספר שלם	גבולות הקטע
eps	מספר ממשי	גודל השגיאה המותרת: $10^{-3} = 0.001$
n	מספר שלם	מספר החלקים של הקטע. n זוגי
S ₀	מספר ממשי	השטח מחישוב קודם
S ₁	מספר ממשי	השטח מחישוב נוכחי
w	מספר ממשי	רוחב הרצועה: $(b-a)/(3*n)$
c	מספר ממשי	שיעור x _i - ערך ה-x הראשון בשלשה

(1) אתחול נתונים: $a \leftarrow 1, b \leftarrow 5, n \leftarrow 2, eps \leftarrow 0.001$

(2) $S_0 \leftarrow 1, S_1 \leftarrow 0$

(3) כל עוד $|S_0 - S_1| > eps$ בצע:

(3.1) $S_0 \leftarrow S_1$

(3.2) $n \leftarrow n * 2$

(3.3) חישוב-שטח $S_1 \leftarrow (a, b, n)$

בהתחלה נקבע n להיות 2 כדי שהחישוב הראשון יתחיל עם 4 רצועות.

F(x) פעולה המחזירה את x^4

חישוב-מקטע (x, y, z, w)

{ פעולה המקבלת את שלשת נקודות x, y ו-z ואת רוחב הרצועה w ומחזירה את השטח עבור נקודות אלו. הנחה: $a \leq x < y < z \leq b$ }

החזר: $w * (F(x) + 4 * F(y) + F(z))$

חישוב-שטח (a, b, n)

{ פעולה המחשבת את השטח תחת גרף הפונקציה בתחום [a, b] עבור n הנוכחי }

(1) $w \leftarrow (b - a) / (3 * n)$

(2) $c \leftarrow a$

(3) $S \leftarrow 0$

(3) עבור i מ-1 ועד מחצית n בצע:

(3.1) חישוב-מקטע $S \leftarrow S + (c, c+w, c+2w)$

(3.2) $c \leftarrow c + 2w$

(4) החזר את S

פתרון בשפת C נכתב על ידי זיוה קונצמן

$$S=S[1,3]+S[3,5]$$

$$f(x)=x^4$$

$$S=4/12(1^4+4*2^4+3^4)+4/12(3^4+4*4^4+5^4)=1/3(1+64+81)+1/3(81+1024+625)=$$

$$146/3+1730/3=1876/3=625\frac{1}{3}$$

.ב

```
#include<stdio.h>
#define a 1
#define b 5
#define diyuk 0.001

float f(float x)
{
    f=sqr(x)* sqr(x);
}

// חישוב שטח של מקטע אחד
float area1(float x1,float x2,float x3,int n)
{
    return (b-a)/(3*n) * (f(x1) + 4*f(x2) + f(x3));
}

void main()
{
    float x1, x2, x3, c, h, s1=0, s2=0;
    int n = 4, i;
    h = (b-a)/ (3*n);
    c = a;
    for (i = 1 ; i <= n/2 ; i++)
    {
        s1 += area1 (c, c+h, c+2*h, n);
        c = c + 2*h;
    }
    n *= 2;
    h = (b-a)/ (3*n);
    c = a;
    for ( i = 1 ; i <= n/2 ; i++)
    {
        s2 += area1(c, c+h, c+2*h, n);
        c = c + 2*h;
    }
    while (abs(s2-s1) > diyuk)
    {
```

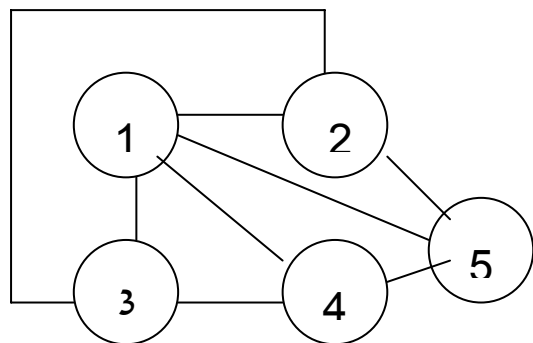
```

s1 = s2;
s2 = 0;
c = a;
n *= 2;
for (i = 1 ; i <= n/2 ; i++)
{
    s2 += area1(c, c+h, c+2*h, n);
    c = c + 2*h;
}
}
printf ("%f", s2);
}
    
```

הפתרון לשאלה זו נכתב ע"י: זיוה קונצמן

שאלה 10:

א.



ב. מספר הקשתות בגרף "גלגל" W_n הוא $2n-2$.

הסבר: מצומת 1 תמיד יוצאות $n-1$ קשתות. מצומת 2 תמיד יוצאת עוד קשת אחת לצומת n . מצומת 3 תמיד יוצאות עוד 2 קשתות (לצומת 2 ולצומת 4). כל שאר הצמתים - - מצומת 3 עד צומת n , לא כולל צומת n - ז.א. $n-3$ צמתים, $n-4$ צמתים - מכל צומת כזו יוצאת עוד קשת אחת, לצומת העוקבת לה.

$$\text{סה"כ: } n-1+1+2+n-4=2n-2$$

ג. לא קיים n בעבורו נוצר מעגל אוילר, מכיוון שתמיד צומת 2 היא בדרגה אי-זוגית - קשורה לצומת 1, לצומת 3 ולצומת n . ומעגל אוילר קיים רק אם כל הצמתים בגרף הן בדרגה זוגית.

ד. המספר המינימלי של קבוצות ב"חלוקה זרה" בגרף "גלגל" W_n תלוי ב n .

אם n זוגי, המספר המינימלי הוא 4, אם n אי-זוגי - המספר המינימלי הוא 3. הסבר: 1 תמיד יהווה קבוצה נפרדת, כי כולם קשורים אליו.

אם n אי-זוגי – כל המספרים הזוגיים יהיו קבוצה שניה וכל המספרים האי-זוגיים יהיו קבוצה שלישית.

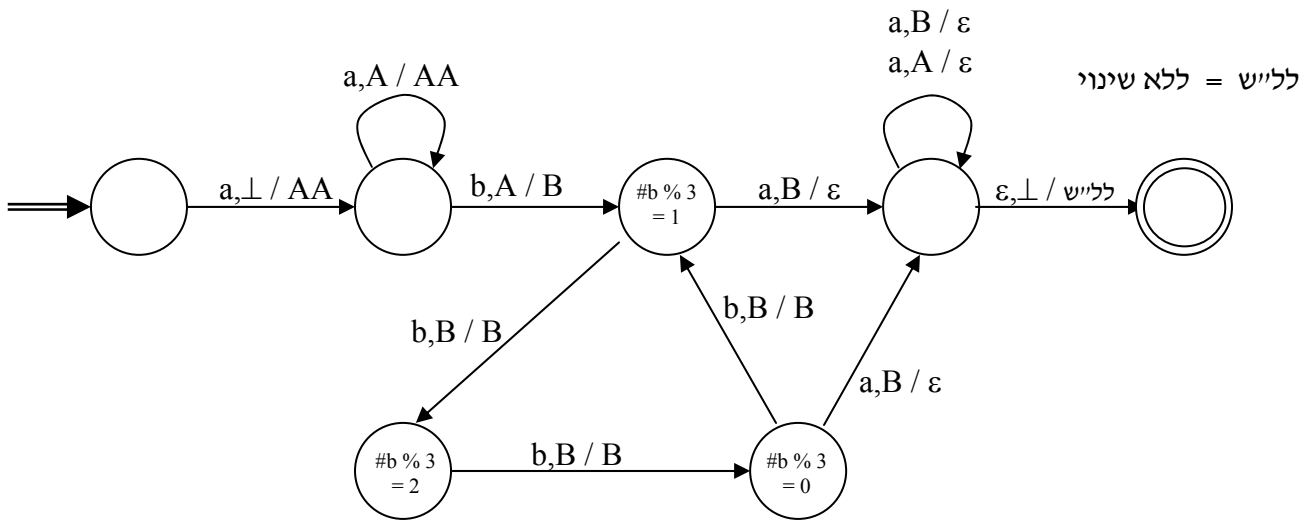
אם n זוגי – שוב, 1 לבד, כל הזוגיים יהיו קבוצה שניה, כל האי-זוגיים יהיו קבוצה שלישית ו- n יהיה קבוצה רביעית.

מכיוון ש-2 קשור תמיד לצומת n , אם n זוגי, הוא לא יכול להיות עם קבוצת הזוגיים ואז נוססת הקבוצה הרביעית.

שאלה 11:

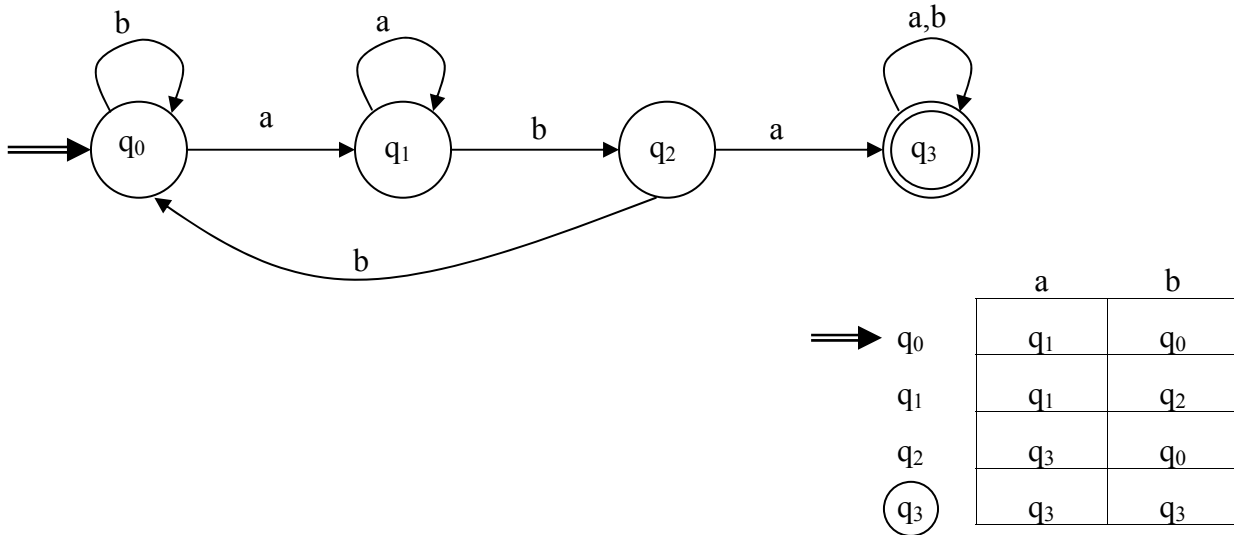
תנאים: $m > 0, k > 0$
 $m \bmod 3 \neq 2$ (שארית = %)
 כלומר - מספר ה- b יהיה: $3j$ או $3j + 1$
 $L = \{d^k b^m a^{2k+m}\}$
 $j = 0, 1, 2, \dots$

- א. המילה הקצרה ביותר: $m = 1, k = 1$ abaaa
- ב. אוטומט מחסנית:

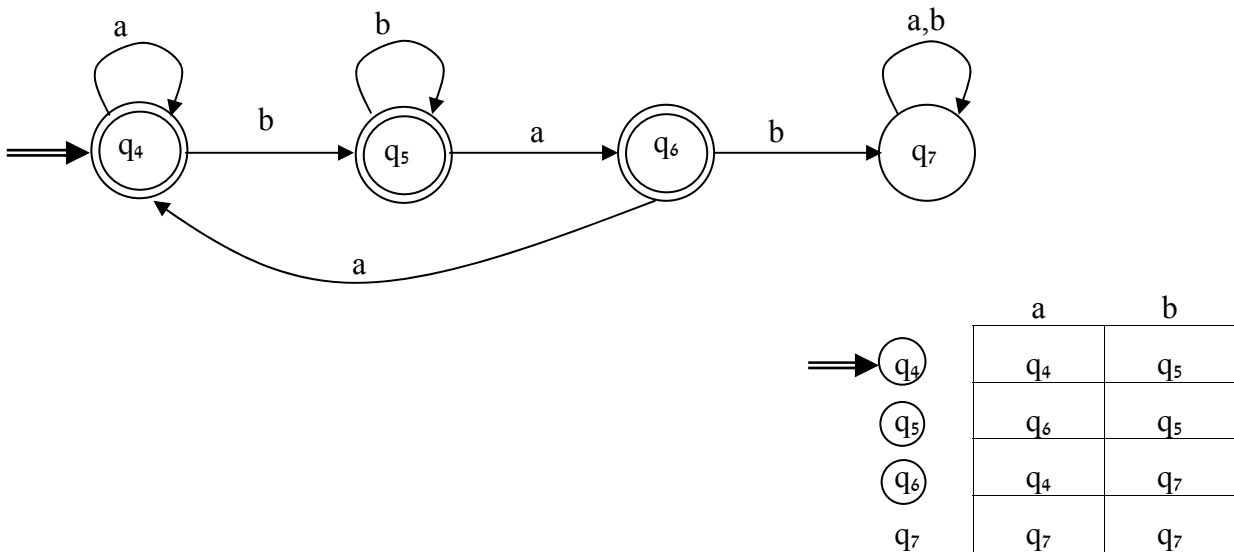


שאלה 12:

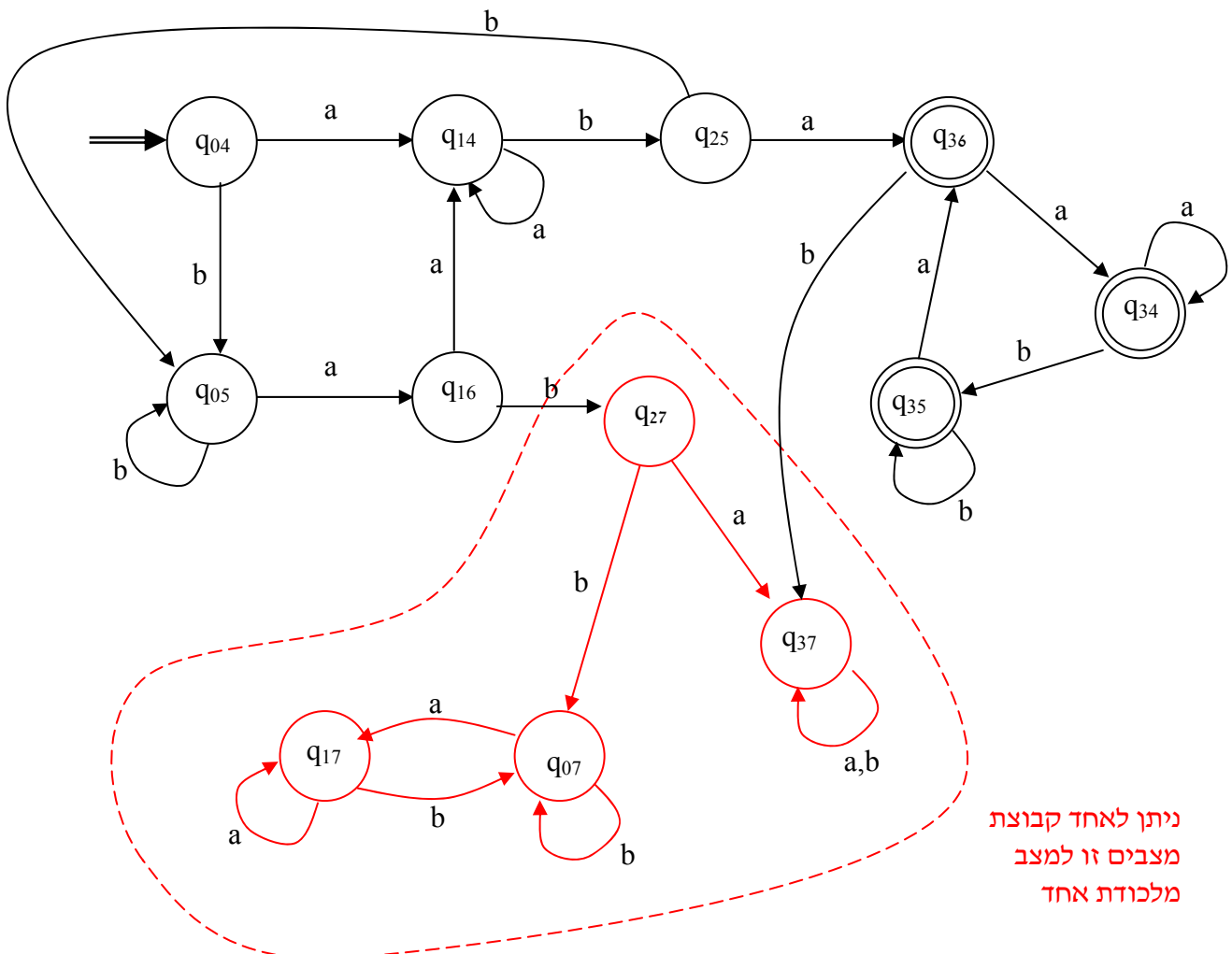
א. אסי"ד המקבל את כל המילים מערך $\{a, b\}$ שיש בהן רצף aba



ב. אסי"ד המקבל את כל המילים מערך $\{a, b\}$ שאינן בהן רצף bab



	a	b
⇒ q ₀₄	q ₁₄	q ₀₅
q ₁₄	q ₁₄	q ₂₅
q ₀₅	q ₁₆	q ₀₅
q ₂₅	q ₃₆	q ₀₅
q ₁₆	q ₁₄	q ₂₇
⊙ q ₃₆	q ₃₄	q ₃₇
q ₂₇	q ₃₇	q ₀₇
⊙ q ₃₄	q ₃₄	q ₃₅
q ₃₇	q ₃₇	q ₃₇
q ₀₇	q ₁₇	q ₀₇
⊙ q ₃₅	q ₃₆	q ₃₅
q ₁₇	q ₁₇	q ₂₇



פרק ב'

מודלים חישוביים

הפתרון לפרק זה נכתב ע"י רחל לודמר.

תרגיל 13:

א.

	השפה	סעיף
$L_1 = \{1^n \mid n \geq 0\}$	$L = \{0^n \mid n \geq 0\}$	(i)
$L_2 = \{R(0^n \mid n \geq 0)\} = \{0^n \mid n \geq 0\}$		
$L_3 = \{0^n 1^n \mid n \geq 0\}$		
$L_1 = \{1\tilde{w}1 \mid w \in (a,b)\}$	$L = \{0w0 \mid (a,b) \text{ מעל } w\}$	(ii)
$L_2 = \{R(0w0) \mid w \in (a,b)\} = \{0R(w)0 \mid w \in (a,b)\}$		
$L_3 = \{0w0 \cdot 1\tilde{w}1 \mid w \in (a,b)\}$		
$L_1 = \{1^n (10)^k \mid n, k \geq 0\}$	$L = \{0^n (01)^k \mid n, k \geq 0\}$	(iii)
$L_2 = \{R(0^n (01)^k) \mid n, k \geq 0\} = \{0^n (10)^k \mid n, k \geq 0\}$		
$L_3 = \{0^n (01)^k 1^n (10)^k \mid n, k \geq 0\}$		

ב.

(i) שפה L לא ריקה שעבורה מתקיים $L=L_1$:

$$L = \{0^n, 1^n \mid n \geq 0\}$$

$$= \{\varepsilon, 0, 1, 00, 11, 000, 111, \dots\} = \{\varepsilon, 1, 0, 11, 00, 111, 000, \dots\} = L_1$$

(ii) שפה L לא ריקה שעבורה מתקיים $L_1=L_2$ וגם $L \neq L_1$:

$$L = \{1^n 0^n \mid n > 0\}$$

$$L_1 = L_2 = \{0^n 1^n \mid n > 0\}$$

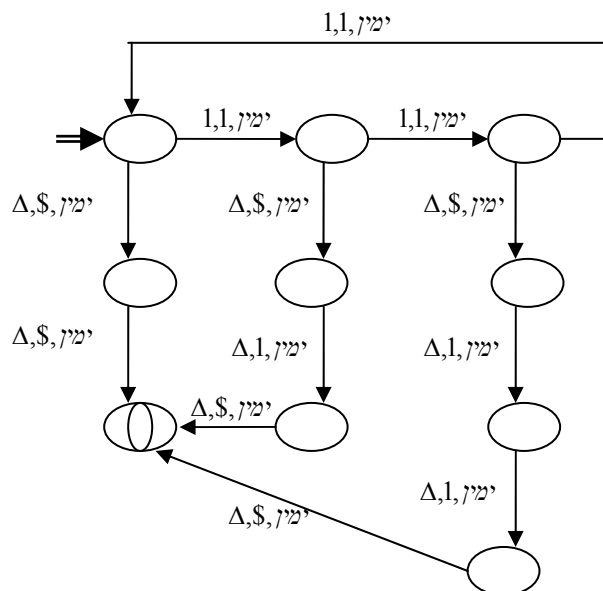
תרגיל 14:

- א. המילה $abba$ שייכת ל- $L_1 \cdot R(L_1)$ ($n=1$), ולא שייכת ל- L_5 ($n=m=1 > 0$), אבל לא מתקיים ($n > m$)
 ב. המילה aba שייכת ל- L_4 ($1 \neq 0$), ולא שייכת ל- L_5 ($n=1, m=0$) בניגוד להגדרת השפה.
 ג. המילה $a^3b^6b^4$ שייכת ל- $L_2R(L_2)$: הרישא a^3b^6 ששייכת ל- L_2 , $6 \neq 3$, הסיפא b^4 שייכת ל- $R(L_2)$
 $n=0, m=4$. מילה זו $a^3b^6b^4$ אינה שייכת ל- L_5 , המילה אינה מסתיימת ב- a בודד ובתחילתה יש יותר מ a בודד.
 ד. המילה הריקה נמצאת בשפה L_3 ($n=m=0$), אך לא נמצאת ב- L_2 (דרוש $n \neq m$). לכן גם לא תמצא בשפת ההיפוך ולא בשרשורם.
 ה. הטענה $L_5 = R(L_5)$ נכונה. הנימוק מוכח באופן מתמטי.

$$\begin{aligned} R(L_5) &= \{ab^m b^n a \mid n, m > 0, n > m\} \\ &= \{ab^{m+n} a \mid n, m > 0, n > m\} \\ &= \{ab^{n+m} a \mid n, m > 0, n > m\} \\ &= \{ab^n b^m a \mid n, m > 0, n > m\} = L_5 \end{aligned}$$

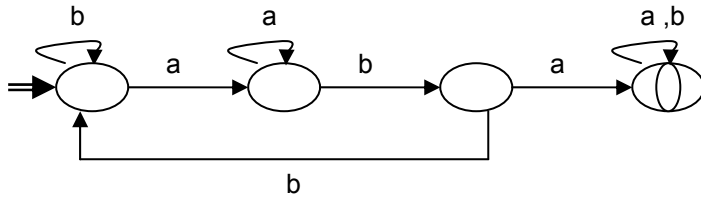
תרגיל 15:

מכונת טיורינג המחשבת את הפונקציה $f(x) = x \bmod 3 \mid x \geq 0$, תוצאת הפונקציה תיכתב על הסרט במוסכמות המקובלות של המכונה.

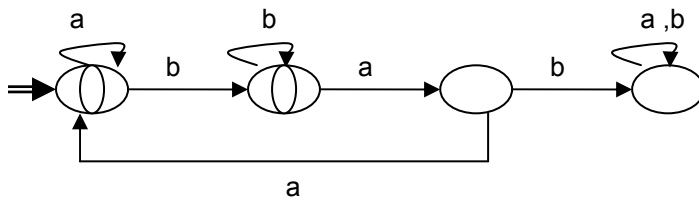


שאלה 16:

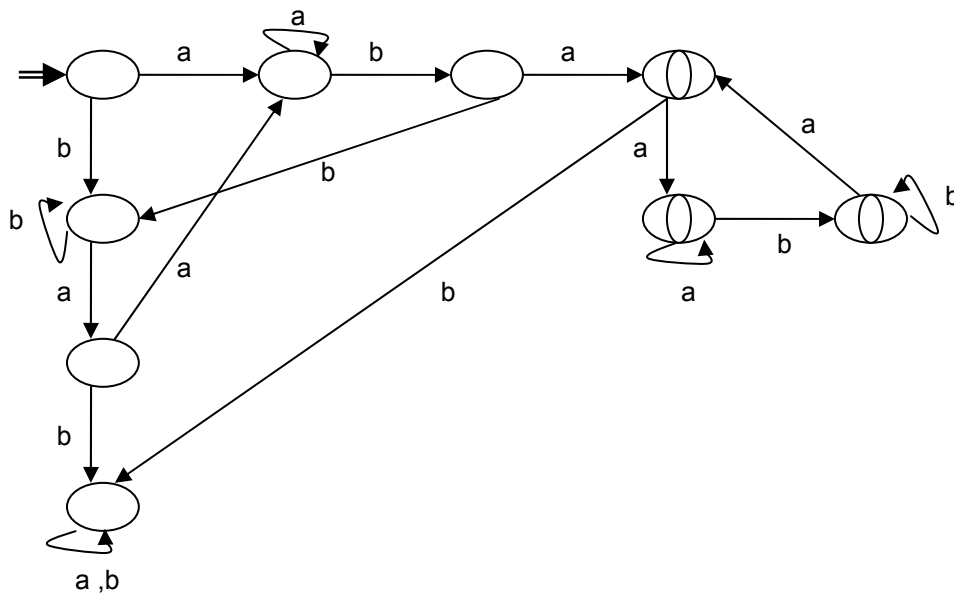
א. אסייד מעל $\{a,b\}$ המקבל את כל המילים שיש בהם את הרצף aba .



ב. אסייד מעל $\{a,b\}$ המקבל את כל המילים שאין בהם את הרצף bab .



ג. אסייד מעל $\{a,b\}$ המקבל את כל המילים שיש בהם את הרצף aba , ואין בהם את הרצף bab . ניתן לבנות אוטומט "מכפלה" משני התנאים או לבנות את האוטומט באופן הרגיל.



פרק ב'

תכנות מונחה עצמים Java

הפתרון לפרק זה נכתב ע"י ...

תראו 17:

תראו 18:

תראו 19:

לאוה 20:

פרק ב'תכנות מונחה עצמים C#תרגיל 21:

הפתרון נכתב על ידי סרחאן תאיר

כל סעיפי השאלה פתורים ביחד ומשולבים בתוכנית שלמה.
הפרוייקט מחולק ל 3 מחלקות עיקריות מחלקה ששומרת מידע עבור התשובות .
ומחלקה אשר שומרת מידע עבור הסקר עצמו כוללת את השאלה ואת תאריך פרסום הסקר ואת המידע
עבוד כל התשובות של המשתתפים .
ומחלקה ששומרת את המידע על 50 הסקרים החדשים היא מוסיפה סקר חדש ושומרת שהסקרים
החדשים יהיו רק 50 הסקרים החדשים ביותר.
הרעיון: מערך של אובייקטים מסוג סקר וכל סקר הוא בעצם מכיל את התשובות של המשתתפים .

// המחלקה מגדירה את אפשרויות התשובה של שאלת הסקר

```
using system;
using system.collections.generic;
using system.text;
class answer
{
    // מספר התשובות של שאלת הסקר
    private const int n = 4 ;

    // מספר המשתתפים שעונים על שאלת הסקר
    public int[] answers;

    //בנאי לאתחול מספר המשתתפים לכל תשובה
    public answer()
    {
        this.answers = new int[n];
    }

    // פעולה שקולטת את הבחירה של התשובה
    // ומעדכנת את נתוני מספר המשתתפים
    public void inputanswer()
    {
        console.WriteLine("enter your answer. between 1..4");
        int personanswer = int.Parse(console.ReadLine());
        this.answers[personanswer - 1]++;
    }
}
```

```
// פעולה המחזירה את נתוני התשובות לסקר כמחרוזת
public override string toString()
{
    int max=0;
    for (int ans = 0; ans < answers.length; ans++)
    if (answers[ans] > max) max = answers[ans];
    return "select 1 answer= " + answers[0] +
           "select 2 answer= " + answers[1] +
           "select 3 answer= " + answers[2] +
           "select 4 answer= " + answers[3]+
           "the max person is=> "+max;
}
}
```

```
// המחלקה מגדירה סקר אשר שומר את תאריך פרסום הסקר,
// את שאלת הסקר ואת התשובות של כל המשתתפים.
class survey
{
    private string surveydate; // תאריך פרסום הסקר
    private string questionsurvey; // שאלת הסקר
    public answer surveyanswer; // מספר המשתתפים ותשובותיהם

    public survey(answer a) // אתחול התשובות לסקר
    {
        this.surveyanswer = a;
        this.surveydate = "0/00/0000";
        this.questionsurvey = "its empty survey";
    }

    public void setsurveydate(string s)
    {
        this.surveydate = s;
    }

    public string getsurveydate()
    {
        return this.surveydate;
    }

    public void setquestionsurvey(string s)
    {
        this.questionsurvey = s;
    }
}
```

```
public string getquestionsurvey()
{
    return this.questionsurvey;
}

public override string toString()
{
    return this.surveyanswer.toString();
}

// מספר כל המשתתפים בסקר מסויים
public int numberofparticipant()
{
    int sum = 0;
    for (int i = 0; i < this.surveyanswer.answers.length; i++)
        sum += this.surveyanswer.answers[i];
    return sum;
}
}
```

```
using system;
using system.collections.generic;
using system.text;

// מחלקה אשר מגדירה את מאגר 50 הסקרים האחרונים באתר "הסקר השבועי"
class allsurvey
{
    public survey[] theweeksurvey; // מערך של 50 הסקרים האחרונים
    private const int numberofsurvey = 50; // מספר הסקרים האחרונים

    // אתחול מערך כל הסקרים
    private int currentsurvey=0; // מספר הסקרים הנוכחי
    public allsurvey()
    {
        this.theweeksurvey = new survey[numberofsurvey];
    }

    // פעולה המקבלת סקר ומוסיפה אותו למאגר נתוני כל הסקרים
    public void add(survey s)
    {
        if (currentsurvey < this.theweeksurvey.length -1 && currentsurvey > 0)
        {
```

```
// הזות כל הסקרים מקום אחד שמאלה
for (int k = currentsurvey; k > 0; k--)
    this.theweeksurvey[k + 1] = this.theweeksurvey[k];
    currentsurvey++;
}
else
    currentsurvey = 0;
this.theweeksurvey[0] = s; // הכנסת הסקר החדש למקום הראשון ברשימת
                           הסקרים
// קליטת תאריך פרסום הסקר והדפסת: מספר המשתתפים, תשובותיהם
// ואת מספר המשתתפים המקסימלי.
public void inputsurveydate()
{
    console.WriteLine("enter the survey date....");
    string s=console.ReadLine();
    for (int seker = 0; seker < this.theweeksurvey.Length; seker++)
    {
        survey sur = this.theweeksurvey[0];
        if (s == sur.getsurveydate())
            console.WriteLine(sur.surveyanswer.ToString());
    }
}
// פעולה המחזירה את מספר הסקרים שבהם יש יותר מ- 1000 משתתפים
public int numberofsurveybegerthantha()
{
    int countsurvey=0; // מספר הסקרים
    // סריקת מערך הסקרים
    for (int seker = 0; seker < this.theweeksurvey.Length; seker++)
        // אם קיים סקר שבו השתתפו יותר מ- 1000 משתתפים
        if (this.theweeksurvey[seker].numberofparticipant() > 1000)
            countsurvey++;
    return countsurvey;
}
}
```

```

using system;
using system.collections.generic;
using system.text;

namespace weeksurvey
{
    class program
    {
        // דוגמא פשוטה לשימוש בפעולות המחלקות
        static void main(string[] args)
        {
            answer a = new answer();
            a.inputanswer();
            survey s = new survey(a);
            allsurvey all = new allsurvey();
            all.add(s);

            for (int i = 0; i < all.theweeksurvey.length; i++)
            {
                if (all.theweeksurvey[i] != null)
                    console.writeline(all.theweeksurvey[i]);
            }

            console.readline();
        }
    }
}

```

תראו 22:

תראו 23:

לאזה 24: