

שאלות שחוברו ע"י משתתפי קורס "שאלות מבחן" תשע"ג
שאלות ל"עיצוב תוכנה"

תוכן עניינים

- 2 שאלות שחוברו ע"י יהודה אבני
- 3 שאלות שחוברו ע"י דורית בן דוד
- 4 שאלות שחוברו ע"י אוי גרינולד
- 12 שאלות שחוברו ע"י טלי דרור
- 15 שאלות שחוברו ע"י חני טוראל
- 16 שאלות שחוברו ע"י וג'יה כבהה
- 17 שאלות שחוברו ע"י רחל לודמר
- 21 שאלות שחוברו ע"י דפנה לוי-רשתי
- 27 שאלות שחוברו ע"י רחלי צרניחוב
- 29 שאלות שחוברו ע"י דני קשת

שאלות שחוברו ע"י יהודה אבני

שאלה 1 – שאלת מעקב רקורסיבי (Backtracking) למבחן מתכונת ב"עיצוב תוכנה" – יהודה אבני

נתונה הפעולה הרקורסיבית הבאה:

```
static int What(int[,] matrix, int i, int j)
{
    int N = matrix.GetLength(0);
    if (matrix[i, j] == 0) return 0;
    if (matrix[i, j] == 2) return 0;
    matrix[i, j] = 2;
    if (i == N - 1 && j == N - 1) return 1;
    if ((i < N - 1) && What (matrix, i + 1, j) == 1 ||
        ((i > 0) && What (matrix, i - 1, j) == 1) ||
        ((j < N - 1) && What (matrix, i, j + 1) == 1) ||
        ((j > 0) && What (matrix, i, j - 1) == 1))
        return 1;
    matrix[i, j] = 1;
    return 0;
}
```

הפעולה מקבלת מטריצה matrix (מספר שורות שווה למספר עמודות), i ו-j הם אינדקס של איבר במטריצה.

1 נתונה מטריצה

```
int[,] matrix = {{1,1,0,1},
                 {0,1,1,1},
                 {1,0,0,1},
                 {0,1,1,1}};
```

מה תחזיר הפעולה `What(matrix,0, 0)` ?

2 נתונה המטריצה

```
int[,] matrix = {{1,1,0,1},
                 {0,1,1,1},
                 {1,0,0,0},
                 {0,1,1,1}};
```

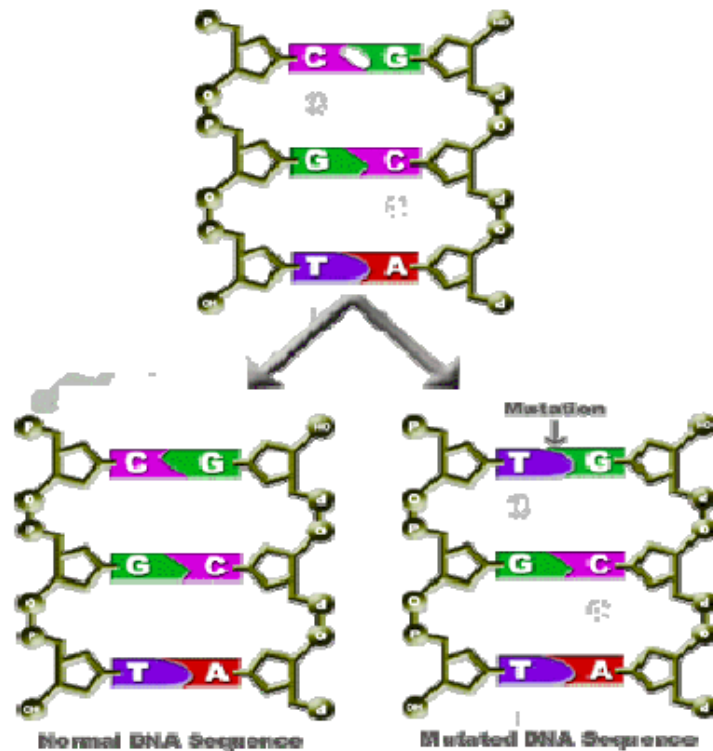
מה תחזיר הפעולה `What(matrix,0, 0)` ?

כתוב במילים שלך מה מבצעת הרקורסיה.

שאלות שחוברו ע"י דורית בן דוד

שאלה 1 – שאלה למבחן מתכונת ב"עיצוב תוכנה" – דורית בן-דוד

לצורך פיתוח תוכנה לביצוע הדמיה של תהליכים גנטיים ב-DNA, נכיר תחילה את מבנה ה-DNA: הדנ"א בנוי משני גדילים מסולסלים הנמשכים בכיוונים מנוגדים ואלה מתקשרים זה לזה על-ידי קשרים בין "זוגות בסיסים": ציטוזין (C) עם גואנין (G), ואדינין (A) עם תימין (T). הסליל הכפול מתפצל ובעזרת אנזימים נוצר מיד בן-זוג הולם לכל אחד משני הגדילים. בסופו של דבר מתקבלים שני סלילים כפולים. (ראה תרשים) לאחר תהליך השכפול נוצרים שני דנ"א, כאשר חצי סליל הוא מהדנ"א המקורי והחצי המשלים נוצר בשיכפול. אם הכל מצליח מקבלים דנ"א תאומים זהים. לפעמים תהליך השכפול משתבש ואז נוצרת מוטציה. דנ"א מיוצג על ידי שתי רשימות מקושרות המכילות תווים (A,G,T,C)



א. כתוב את כותרת המחלקה Dna ואת התכונות שלה על פי הייצוג המתואר.

ב. ממש את הפעולות הבאות במחלקה

הפעולה הבודקת האם הדנ"א תקין. תקין משמעותו שהוא מכיל רק את האותיות הנכונות ושהזוגות תואמים. הפעולה מחזירה אמת אם תקין ושקר אחרת.	CheckNormal()
הפעולה מחזירה דנ"א חדש המכיל מחצית מהדנ"א הישן ומחצית חדשה של דנ"א. לאחר ביצוע הפעולה יש לנו שני דנ"א הדומים במבנה שלהם.	Duplicat()

יש להקפיד לבצע את פעולת השכפול כפי שהדבר מבוצע בטבע. כלומר, שיכפול מלא.

שאלות שחוברו ע"י אוי גרינולד

שאלה 1 – הרגלי הכביסה במשפחות לוי וכהן – שאלת סיפור למבחן מתכונת ב"עיצוב תוכנה" – אוי
בבית משפחת לוי מניחים בארון הבגדים ערימות על פי סוג פריט הבגד: גופיות בערימה, תחתונים בערימה, חולצות בערימה וכך הלאה. כאשר חוזרים הבגדים מהכביסה כל בגד מונח בראש הערימה שלו.
בבית משפחת כהן מסדרים את הבגדים גם בערימות על פי סוג הבגד, אלא שכדי שלא ילבשו כל הזמן את אותו הבגד וכך יתבלה מהר יותר, אחרי הכביסה שמים כל בגד בתחתית הערימה שלו.
בשתי המשפחות מוציאים בגדים נקיים מהארון כל פריט מראש הערימה שלו.
עבור אותם בגדים בדיוק, מעוניינים לדעת את סדר לבישת הבגדים בכל אחת מהמשפחות בשבוע האחרון. יש להתייחס רק לפריטים הבאים: גופיות, תחתונים, חולצות ומכנסים.
כלומר: הנח שבבית משפחת לוי ובבית משפחת כהן לובשים את אותם הבגדים אלא שכל אחת מהמשפחות נוהגת להחזיר את בגדיה אחרי כביסה לארון על פי הסדר שהוזכר לעיל.
עבור כל פריט לבוש **Item** יישמר המידע הבא: מספר סידורי של הפריט כאשר המספר הסידורי נקבע באופן אוטומטי ומספרו הסידורי מתחיל מ 1. קוד פריט: 1-עבור גופיה, 2-תחתון, 3-חולצה, 4-מכנס, וצבע הפריט.
מחלקת **ביגוד Clothing** שומרת את סדר לבישת פרטי הלבוש בשתי המשפחות. כאשר ההתייחסות היא לאותם הבגדים.

להלן חלק מממשק המחלקה **ביגוד Clothing**

תפקיד הפעולה	הפעולה
פעולה מאחלת ביגוד ריקה מבגדים.	<code>public Clothing()</code>
פעולה המוסיפה פריט לבוש גם לארון הבגדים של משפחת לוי וגם לארון הבגדים של משפחת כהן.	<code>public void add (Item item)</code>
פעולה המקבלת מספר משפחה 1-עבור לוי, 2- עבור כהן ומספר קוד פריט. הפעולה מוציאה את פריט הלבוש מהארון המתאים ומחזירה אותו.	<code>public Item remove(int family, int code)</code>
פעולה המוציאה פריט לבוש לפי קוד מהארונות של משפחות לוי וכהן ומציגה כפלט זוג סדור של מספרי הקוד של הפריטים שהוצאו. למשל: { 3,1 } אומר שפריט בעל קוד 3 הוצא מהארון של משפ' לוי ופריט בעל קוד 1 הוצא מהארון של משפ' כהן.	<code>public void remove(int code)</code>
הפעולה מחזירה אמת אם קיים עוד פריט לבוש בעל קוד code במחלקת הביגוד, שקר- אחרת.	<code>public boolean hasMore(int code)</code>

- 1) כתוב ייצוג למחלקה **Item** . כתוב כותרת, תכונות ותיעוד לכל תכונה.
- 2) כתוב ייצוג למחלקה **Clothing** . כתוב כותרת, תכונות ותיעוד לכל תכונה.
- 3) ממש את הפעולה `void remove(int code)`
- 4) בממשק המחלקה קיימות שתי פעולות `remove` . הסבר תופעה זו.
- 5) כתוב פעולה חיצונית אשר מקבלת עצם מטיפוס `Clothing` מאותחל ומציגה כפלט זוגות של מספרים סידוריים עבור פרטי גופייה. חובה להשתמש בפעולות הממשק

בכל יום 2 המשפחות מוציאות גופייה ותחתונים ופעם בשלושה ימים מוציאים חולצה. לשם פשטות מיד עם הוצאת פריט הכנס אותו בחזרה לארונות של המשפחות בהנחה שהפריט כובס.

- 6) כתוב פעולה חיצונית אשר מקבלת עצם מטיפוס `Clothing` מאותחל ומלא בבגדים. הפעולה תוציא גופייה ותחתונים עבור כל משפחה בכל אחד מימות השבוע וחולצה כל שלושה ימים. מיד עם הוצאת הפריט הוא מוחזר לארונות של המשפחות. לכל ארון על פי הכללים שנהוגים באותה משפחה. חובה להשתמש בפעולות הממשק.

פתרון לשאלה

```
/*
 * Evi's question - question of type adt for matkonet
 * Item is an article of clothing
 */
package Matkonet;

/**
 * @version 5/3/13
 * @author evi
 */
public class Item
{
    private static int counter=0;
    private int code; // item's code 1-undershirt, 2- underwear 3- shirt 4- trouser
    private int serial; // serial number of item
    private String color; // item's color

    public Item(int code,String color)
    {
        Item.counter++;
        this.code = code;
        this.serial = Item.counter;
        this.color = color;
    }

    public int getCode()
    {
```

```
        return code;
    }

    public int getSerial()
    {
        return serial;
    }

    public String getColor()
    {
        return color;
    }

    @Override
    public String toString()
    {
        String str="code:"+this.code + " , serial number:" + this.serial+" , color: " + this.color;
        return str;
    }
}
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package Matkonet;

import unit4.collectionsLib.Queue;
import unit4.collectionsLib.Stack;

/**
 * @version 13/3/13
 * @author evi
 */
public class Clothing
{
    public static final int N=5; // number of types of clothing
    private Stack<Item>[] levi;
    private Queue<Item>[] chohen;

    public Clothing()
    {
        this.levi = new Stack[N];
        for (int k=0; k<this.levi.length; k++)
            this.levi[k] = new Stack<Item>();
        this.chohen = new Queue[N];
        for (int k=0; k<this.chohen.length; k++)
            this.chohen[k] = new Queue<Item>();
    }
}
/**
 * add item to levi's closet and to chohen's closet
```

```
* @param item : Item
*/
public void add (Item item)
{
    int index = item.getCode()-1;
    this.levi[index].push(item);
    this.chohen[index].insert(item);
}
/**
 * removes item of clothing with code is same families
 * @param family: int 1-levi, 2-chohen
 * @param code : int (1-5)
 * @return item in family: Item
 */
public Item remove(int family, int code)
{
    int index = code-1;
    if (family==1)
        return this.levi[index].pop();
    else // family==2
        return this.chohen[index].remove();
}
/**
 * removes item of clothing with code is same families
 * @param code : int (1-4)
 */
public void remove(int code)
{
    int index = code-1;
    System.out.print("{ "+this.levi[index].pop().getSerial()+" ");
    System.out.print(this.chohen[index].remove().getSerial()+"}");
}
/**
 * @param code: int (1-5)
 * @return true if cloth with code exists, false - otherwise
 */
public boolean hasMore(int code)
{
    return !this.levi[code-1].isEmpty() && !this.chohen[code-1].isEmpty() ;
}
@Override
public String toString()
{
    String str="levi:\n";
    for (int k=0;k<this.levi.length; k++)
        str+= this.levi[k]+" \n";
    str+="\nChohen:\n";
    for (int k=0; k<this.chohen.length; k++)
        str+= this.chohen[k]+" \n";
    return str;
}
```

```
    }  
}  
  
/*  
 * Evi's question - question of type adt for matkonet  
 * Item is an article of clothing  
 */  
package Matkonet;  
  
import java.util.Scanner;  
  
/**  
 * @version 5/3/13  
 * @author evi  
 */  
public class MainClothing  
{  
    static Scanner in = new Scanner(System.in);  
    public static Clothing fillCloset()  
    {  
        Clothing cl = new Clothing();  
        int code;  
        String color;  
        Item item;  
        System.out.println("Enter code(1-5)");  
        code = in.nextInt();  
        while (code >= 1 && code <= 5)  
        {  
            System.out.println("Enter color");  
            color = in.next();  
            item = new Item(code,color);  
            cl.add(item);  
            System.out.println("Enter code(1-5)");  
            code = in.nextInt();  
        }  
        return cl;  
    }  
}  
/**  
 * print order of serial numbers in each family  
 * @param cl: Clothing  
 * @param code : int (1-5)  
 */  
public static void order(Clothing cl, int code)  
{  
    while (cl.hasMore(code))  
        cl.remove(code);  
}  
  
public static void useAndClean(Clothing cl)  
{
```



```
Item item1, item2;
for (int d=1; d<=7; d++)
{
    if (cl.hasMore(1))
    {
        item1 = cl.remove(1, 1);
        item2 = cl.remove(2, 1);
        cl.add(item1);
    }
    if (cl.hasMore(2))
    {
        item1 = cl.remove(1, 2);
        item2 = cl.remove(2, 2);
        cl.add(item1);
    }
    if (d%3==0 && cl.hasMore(3))
    {
        item1 = cl.remove(1, 3);
        item2 = cl.remove(2, 3);
        cl.add(item1);
    }
}
}
/**
 * @param cl: Clothing
 * @param family: int 1-levi, 2- chohen
 * @param code: int (1-4)
 * @return
 */
public static int longestColor(Clothing cl, int family, int code)
{
    return 0;
}
public static void main(String[] args)
{
    Clothing closet = fillCloset();
    System.out.println(closet);
    System.out.println("order of unershirt");
    order(closet, 1);
    System.out.println("");
    useAndClean(closet);
    System.out.println("order of shirts");
    order(closet, 3);
    System.out.println("");
    System.out.println(closet);
}
}
```

שאלה 2 – שאלת סיפור למבחן מתכונת – אוי גרינולד

לשאלה יש 3 גרסאות:

- 1) גרסה ראשונה: מותאמת לעיצוב תוכנה. פתרון מאולץ לחלוקה למחלקות. קיים יחס הכלה. בודק מידול נכון.
- 2) גרסה שנייה: מותאמת ליסודות. שימוש בטיפוס קיים Vector או ArrayList. מערכי מונים. (ראו בקובץ השאלות ליסודות)
- 3) גרסה שלישית: מותאמת ליסודות. קלה יותר. לא מתאימה למציאות. שימוש במערך מונים דו-מימדי. (ראו בקובץ השאלות ליסודות)

שאלה 2 – גרסה ראשונה – אוי גרינולד

במסגרת שמירה על כדור הארץ התקיימה פעילות בקרב בני נוער אשר מטרתה להעלות את המודעות לשמירה על איכות הסביבה.

אחת המשימות הייתה:

בפני החניכים הוצגו 4 פריטים: נייר עיתון, בקבוק פלסטיק, פחית שימורים ובטרייה.

החניכים נדרשו בפרק זמן מוגבל למצוא כמה שיותר שימושים לכל מוצר.

החניכים התחלקו ל 5 קבוצות. הקבוצה שמצאה את מספר השימושים הרב ביותר ניצחה במשימה.

כל קבוצה הציגה את השימושים שמצאה עבור כל מוצר.

מעבר לניצחון במשימה, כל קבוצה נדרשה לכתוב לתעד את השימושים השונים שמצאה לכל מוצר.

שימושים שכיחים למוצרים הובאו לידיעה בעלון התנועה.

זן המדריך הציע למחשב את תוצאות הפעילות.

לשם כך הוגדרו הטיפוסים הבאים:

מוצר	Product	המאופיין על ידי קוד מוצר, שם מוצר ורשימת שימושים
קבוצה	Team	מאופיינת על ידי מספר קבוצה, שם הקבוצה ואוסף המוצרים שהוזכרו לעיל
אוסף קבוצות	TeamSet	מכיל את כל חמשת קבוצות החניכים

הפעולות שניתן לבצע במערכת הממוחשבת הן:

- 1) הוספת שימוש למוצר
- 2) בדיקה אם שימוש למוצר כבר קיים או לא.
- 3) מהו המספר הכולל של שימושים של כל המוצרים שמצאה קבוצה מסוימת
- 4) מי הקבוצה או הקבוצות שניצחו במשימה.
- 5) פעולה נוספת אפשרית, אבל משנה את מבנה הנתונים של השאלה:
- 5) לאיזה מוצר נמצאו הכי הרבה שימושים.

שאלות:

- א- כתוב ייצוג למחלקה **מוצר**. רשום כותרת, תכונות ותיעוד.
- ב- כתוב ייצוג למחלקה **קבוצה**. רשום כותרת, תכונות ותיעוד.
- ג- כתוב ייצוג למחלקה **אוסף-קבוצות**. רשום כותרת, תכונות ותיעוד.
- ד- שרטט תרשים מבנה המחלקות.
- ה- עבור כל פעולה מהפעולות שצוינו לעיל רשום את :
כותרת הפעולה, תיעוד שלה, מה הפעולה מקבלת ומה מחזירה ובאיזה מחלקה כתבת אותה.

פתרון (גרסה 1)

ייצוג למחלקה **מוצר**

כותרת המחלקה: **Product**

תכונות:

private int code;	קוד מוצר
private String name;	שם מוצר
private Node<String> utilities;	רשימת שימושים למוצר
private int coutUtilities;	מספר השימושים למוצר

פעולות:

public boolean isUtilityExist (String name)	פעולה המקבלת שם שימוש ומחזירה אמת הם מופיע ברשימת שימושי המוצר שקר-אחרת
public void addUtility (String utility)	הוספת שימוש למוצר במידה ולא מופיע כבר. במידה והשימוש התווסף מעדכנת את מספר השימושים של המוצר.
public int getCoutUtilities ()	מחזירה את מספר השימושים

ייצוג למחלקה **קבוצה**

כותרת המחלקה: **Team**

תכונות:

private int numSet;	מספר קבוצה
private String name;	שם קבוצה
private Product[] products;	אוסף מוצרים

פעולות:

<code>public void addUtilityToProduct</code> (int productCode, String utility)	הוספת שימוש למוצר בעל קוד productCode
<code>public int getTotalNumberUtilities()</code>	מחזירה את מספר השימושים הכולל שמצאה הקבוצה

ייצוג למחלקה אוסף-קבוצה

כותרת המחלקה: TeamSet

תכונות:

אוסף קבוצות	<code>private Team[] set</code>
-------------	---------------------------------

פעולות:

<code>public void add</code> (int teamNum, int productCode, String use)	הוסף לקבוצה בעלת מספר קבוצה teamNum ולמוצר בכל קוד מוצר productCode את השימוש use
<code>public int maxCounter()</code>	מחזירה כמות שימושים מקסימלית
<code>public void showWinners()</code>	מציגה את הקבוצות שניצחו



שאלות שחוברו ע"י טלי דרור

שאלה 1 – שאלת סיפור למבחן מתכונת ב"עיצוב תוכנה"

Date הוא עצם המתאר תאריך ע"י 3 מספרים שלמים (int day, int month, int year)

Blood הוא עצם המתאר מנת דם, המתוארת ע"י:

- סוג הדם (string btype),
- תאריך התרומה (Date day)
- ומספר מזהה של המנה (string id).

BloodBank הוא עצם המתאר את מאגר מנות הדם הכשירות בבנק הדם (מנת דם יכולה להישמר לכל היותר

30 יום. לאחר היום ה-30 המנה מושמדת ויוצאת מהמאגר).

לפניך ממשק חלקי למחלקה BloodBank

תאור הפעולה	כותרת הפעולה
פעולה הבונה מאגר ריק	public BloodBank ()
פעולה המוציאה מהמאגר את המנות הלא כשירות (של היום ה-30)	public void OutOfBank()
פעולה המקבלת תאריך שהוא תאריך של היום האחרון במאגר (תאריך של המנות הכי חדשות) הפעולה תקלוט את נתוני המנות של יום זה, לכל מנה את הסוג והמזהה שלה, סיום הקלט יהיה שהסוג יהיה "xx"	public void AddBloodForLastDay(Date d)
פעולה המקבלת קוד מזהה של מנה ומחזירה את סוג הדם של המנה (אם המנה לא קיימת הפעולה תחזיר null)	public string FindTheBlood (string id)
פעולה המקבלת סוג דם ומספר x ומוציאה את x המנות הותיקות מסוג זה מהמאגר ומחזירה רשימה של מנות אלו. (אם אין x מנות יחזרו כמה מנות שיש)	public List<Blood> FindXBloodAndRemove (string bt, int x)

1. כתוב את כותרת המחלקה ותכונות המחלקה BloodBank

2. ממש את פעולות הממשק, לכל פעולה ציין גם את יעילות הביצוע של הפעולה שכתבת.

הנח כי למחלקות Blood, Date קיימות הפעולות: בונה, Get/Set לכל תכונה

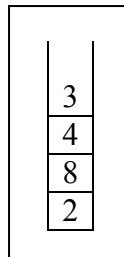
שאלה 2 – שאלת מעקב רקורסיבי למבחן מתכונת ב"עיצוב תוכנה"

נתונות הפעולות הבאות:

```
static int Help(Stack<int> st)
{
    int x,c;
    if (st.IsEmpty() == true) return 0;
    x = st.Pop();
    c = 1 + Help(st);
    st.Push(x);
    return c;
}

static int Mystery (Stack<int> st, int n)
{
    // הפעולה מקבלת מחסנית לא ריקה
    int x = st.Pop(), sum;
    if (n == 1) sum=x;
    else sum = Mystery (st, n - 1) + x;
    st.Push(x);
    return sum;
}
```

```
}  
  
static void Make(Queue<int> qu, int k)  
{  
    if (qu.IsEmpty()) qu.Insert(k);  
    else  
    {  
        int x = qu.Remove();  
        Make(qu, k);  
        qu.Insert(x);  
    }  
}  
  
static void What(Stack<int> st, int z, Queue<int> qu)  
{  
    int p;  
    if (z>0)  
    {  
        p = Mystery (st, z);  
        Make(qu, p);  
        What(st, z - 1, qu);  
    }  
}
```

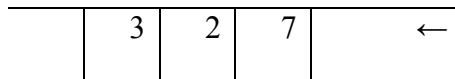


נתונה המחסנית st:

(1) מה תחזיר הפעולה **Help** עבור המחסנית st ?
(תשובה – 4) יש להראות מעקב.

(2) מה תחזיר הפעולה **Mystery** עבור המחסנית st ו-n=3 ?
(תשובה – 15, סכום 3 אברים עליונים במחסנית)

(3) נתון התור qu:



כיצד יראה התור לאחר זימון הפעולה **Make** עבור התור qu ו-K=5.
(תשובה - [5,7,2,3])

(4) נתון התור הריק qu והזימון בפעולה הראשית:

```
What(st, Help(st), qu);
```

עקוב אחר ביצוע הפעולה והראה איך יראו התור והמחסנית בתום ביצוע. אין צורך להראות מעקב על

הפעולות Mystery ו- Make.

(תשובה: המחסנית נשארת ללא שינוי, התור יהיה [3,15,17,7])

שאלות שחברו ע"י חני טוראל

שאלה 1 – שאלה למבחן מתכונת

במשרדי המועצה המקומית אפרתה יש רישום של זוגות צעירים המעוניינים לגור בקרוואן בגבעת התמר.

'זוג צעיר' הוא ישות המיוצגת על ידי המחלקה Fam, ובה שלש תכונות:

1. name מטיפוס String, שם המשפחה של בני הזוג.

2. salary מטיפוס שלם, סך השכר החדשי של שני בני הזוג.

3. date מטיפוס Date, תאריך הרישום לקרוואן.

אנו מניחים שעומדת לרשותנו המחלקה Date, ובה התכונה תאריך, וכן אנו מניחים שיש במחלקה Date פעולה

בשם compareDates המקבלת כפרמטר תאריך, ומחזירה את הערך 1 אם התאריך שהתקבל מאוחר לתאריך

הנוכחי ('הנוכחי' = התאריך של העצם), ואת הערך 1- אם התאריך שהתקבל מוקדם או זהה לתאריך הנוכחי.

במועצה מנהלים את הרישום על ידי שלשה תורים, ואלו הם:

- תור א', ובו זוגות צעירים שלפחות אחד מהם בן להורים תושבי אפרת.
- תור ב', ובו זוגות צעירים שלפחות אחד מהם עולה חדש.
- תור ג', ובו שאר הנרשמים.

כל זוג שנרשם, מוכנס לאחד משלש התורים האלה (כמובן, לסוף אותו תור). מאליו ברור שבפועל, כל אחד מהתורים האלה ממוין לפי תכונת התאריך.

שלושת התורים האלה מיוצגים כשלש תכונות: q1, q2, q3, במחלקה Tamar.

א. כתוב במחלקה Tamar פעולה בשם:

```
insertIntoSortedQueue(Queue<Fam> queue, Fam fam)
```

הפעולה הזו מקבלת שני פרמטרים:

פרמטר ראשון – תור של זוגות צעירים.

פרמטר שני – זוג צעיר.

הפעולה מכניסה את הזוג שקיבלה לתוך התור queue על פי המקום הראוי לו בסדר המיון, ובהנחה שהתור

queue ממוין לפי התכונה date. הפעולה הזו אינה רשאית להניח שהמקום הראוי לזוג על פי סדר המיון,

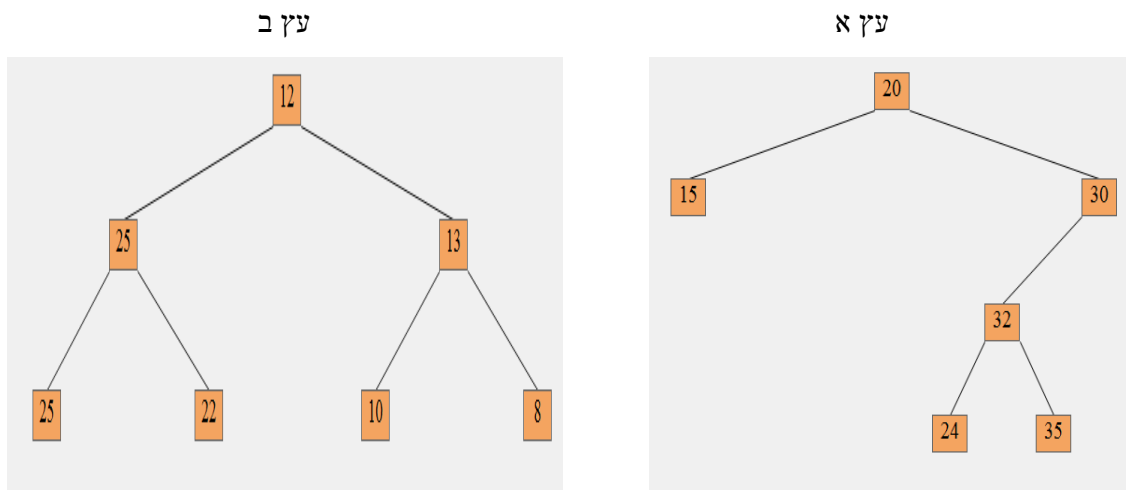
הוא דווקא בסוף התור.

ב. העזר בפעולה שכתבת בסעיף א', וכתוב במחלקה Tamar פעולה בשם reorganize המארגנת מחדש את כל הזוגות הצעירים שנרשמו למגורים בגבעת התמר, כך שבתור א' יהיו אלה שהכנסתם החדשית פחות מ 5,000 ש', ובתור ב' יהיו אלה שהכנסתם החדשית המשותפת גדולה או שווה 5,000 ש'. תור ג' יישאר ריק. באירגון התורים מחדש לא יהיה שום מעמד מיוחד לא לבני תושבי אפרת ולא לעולים חדשים. סדר הזוגות הצעירים בכל אחד משני התורים יהיה זהה במדויק לסדר בו נרשמו הזוגות למגורים. היינו, כל אחד משני התורים יהיה ממוין לפי תכונת התאריך.

שאלות שחברו ע"י וג'יה כבהה

שאלה 1 – שאלת עצים למבחן מתכונת ב"עיצוב תוכנה"

עץ-מסלול-עולה הוא עלה או עץ בינארי שיש בו לפחות מסלול אחד החל מהשורש עד הקצה, שבו כל צומת קטן מאחד הבנים שלו. לפניך העצים הבינאריים א ו ב.



א. איזה מהם הוא עץ-מסלול-עולה? נמק

ב. לפניך סריקה תחילית (משמאל לימין) של עץ מסלול עולה, צייר את העץ :

5,8,10,7

ג. לפניך סריקה תוכית (משמאל לימין) של עץ מסלול עולה. צייר את העץ :

2,13

ד. כתוב פעולה אשר מקבלת עץ בינארי ומחזירה true אם הוא עץ-מסלול-עולה ו false אחרת.

שאלות שחוברו ע"י רחל לודמר

שאלה 1 – שאלת עצים למבחן מתכונת ב"עיצוב תוכנה"

חלק א. נתונה הפעולה : `public static boolean Leaf(BinTreeNode <Integer> tr)`

הפעולה מחזירה אמת אם הצומת היא עלה ושקר אחרת.

לפניך מספר פעולות הנעזרות בפעולה `Leaf`.

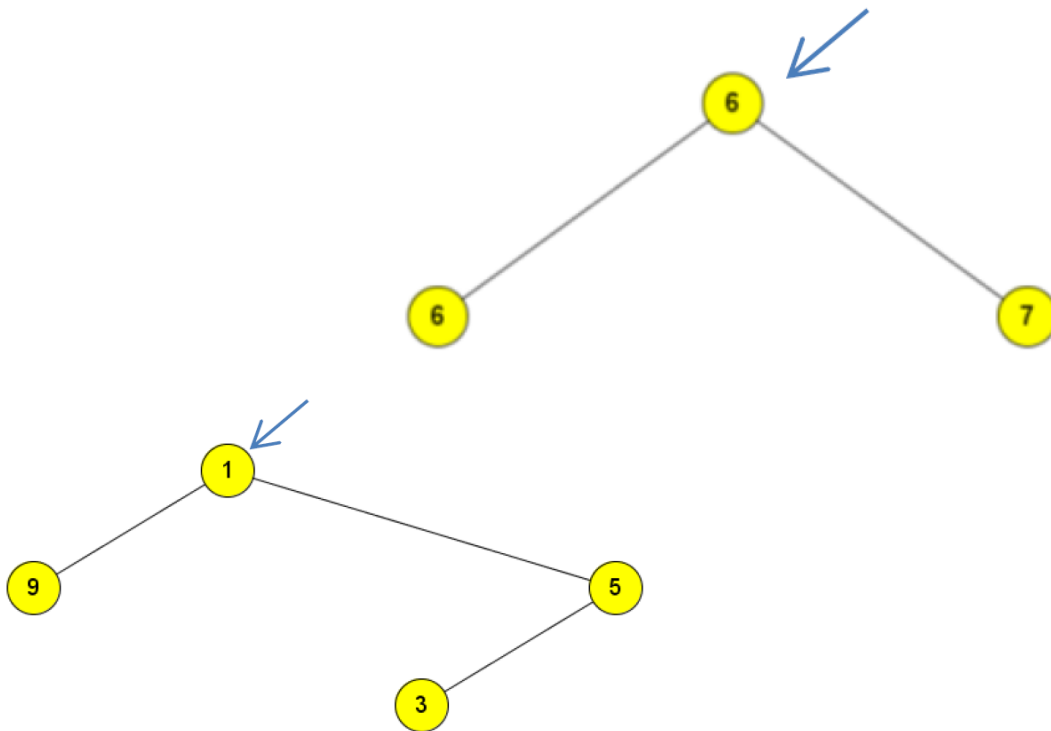
```
public static int sod1(BinTreeNode <Integer> tr, int r)
{
    if( Leaf(tr)) return r;
    if (r%2==0)
    {
        if(tr.getLeft()==null && tr.getRight()!=null)
            return sod1(tr.getRight(),r+1);
    }
    if (r%2==1)
    {
        if(tr.getLeft()!=null && tr.getRight()==null)
            return sod1(tr.getLeft(),r+1);
    }
    return 0;
}
```

```
public static int sod2(BinTreeNode <Integer> tr, int r)
{
    if( Leaf(tr)) return r;
    if (r%2==0)
    {
        if(tr.getLeft()!=null && tr.getRight()==null)
            return sod2(tr.getLeft(),r+1);
    }
    if (r%2==1)
    {
        if(tr.getLeft()==null && tr.getRight()!=null)
            return sod2(tr.getRight(),r+1);
    }
    return 0;
}
```

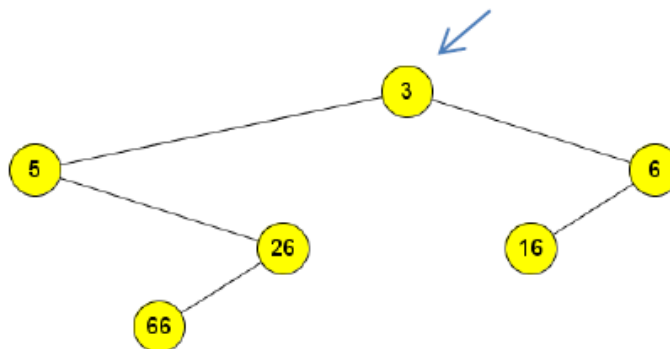
```
public static boolean sodTree(BinTreeNode <Integer> tr)
{
    if (Leaf(tr)) return true;
    if (tr.getLeft()==null || tr.getRight()==null)
        return false;
```

```
if (Leaf(tr.getLeft()) && Leaf(tr.getRight()))  
    return true;  
int x=sod1(tr.getLeft(), 0);  
int y=sod2(tr.getRight(), 0);  
if (x*y==0 )  
    return false;  
return (x==y);  
}
```

א. מה תחזיר הפעולה sodTree עבור העצים הבאים:



ב. רשום עץ מעקב עבור הפעולה sod (כולל מעקב ל sod1, sod2) עבור העץ הבא:



ג. תן דוגמא לעץ עם 9 צמתים לפחות כך שהפעולה sodTree תחזיר true.

ד. תן דוגמא לעץ עם 5 צמתים כך שהפעולה sodTree תחזיר false.

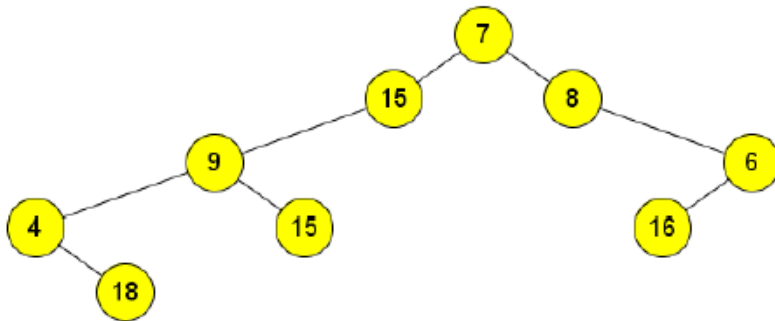
ה. מה מטרת הפעולה.

חלק ב - אופציונלי - אין קשר בין הסעיפים.
כתוב פעולה המקבלת עץ בינרי של תווים. הפעולה תחליף לכל צומת שיש לה שני בנים את תוכנם של שני הבנים.

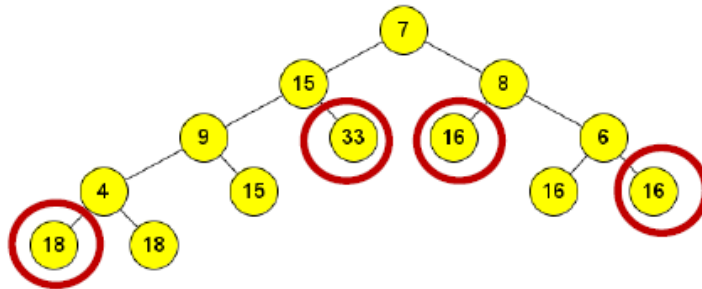
שאלה 2 – שאלת עצים למבחן מתכונת ב"עיצוב תוכנה"

א. כתוב פעולה המקבלת עץ בינרי של מספרים. הפעולה מוסיפה לכל צומת שיש לה בן יחיד, אה (עלה), שערכו הוא סכום העלים של הצאצאים של הצומת.
סדר ההוספה יהיה בסדר של סריקה תחילית של העץ.
לדוגמה

לפני ההוספה



אחרי ההוספה



ב. צייר את העץ שבדוגמא המתקבל לאחר הפעולה שכתבת בסעיף א, לו סדר ההוספה היה בסדר התוכי של סריקת העץ.

שאלה 3 – שאלת סיפור למבחן מתכונת ב"עיצוב תוכנה"

גרסה נוספת שמתאימה ליסודות – ראו בקובץ השאלות של יסודות

מחשבון BMI מאפשר לכם לברר מהו ה-BMI שלכם. מדד מסת הגוף (Body Mass Index) הוא מדד המודד את היחס בין משקל הגוף לגובה. התוצאה מלמדת על היחס (פרופורציה) בין הגובה למשקל של הנבדק. מדד זה משמש כלי למדידת עודף ותת-משקל ועל ידי כך לאבחן התפתחות בעיות בריאות הקשורות לעודף משקל.
כיצד מחשבים את ה-BMI?

BMI = משקל (בקילוגרמים) מחולק בריבוע הגובה (במטרים)

$$BMI = \frac{\text{משקל (בקילוגרמים)}}{(\text{גובה (במטרים)})^2}$$

לדוגמה, אם ה-BMI שלך הוא 23.53 (משקל=57ק"ג, גובה=1.55 מטר) $(57/1.55^2=23.53)$

תוצאה זאת מכניסה אותך לקטגוריה של "משקל בריא" (Healthy Weight). משמעות התוצאה היא שמשקל גופך מקטין משמעותית את הסיכוי שתסבול מבעיות בריאות כגון: מחלות לב, סוכרת, יתר לחץ דם, סוגים מסוימים של סרטן ועוד.

מיקום תוצאת ה-BMI שלך ביחס לקבוצות הייחוס השונות אתה נמצא כאן



טבלה 1

טווח BMI (כולל)	קטגוריית משקל
קטן מ- 18.5	תת משקל
18.5 - 24.9	משקל בריא
25 - 29.9	עודף משקל
30 - 34.9	השמנה
35 - 39.9	השמנת יתר
מ- 40 ומעלה	השמנת יתר חמורה

לצורך שמירת משקל בריא נחשב את צריכת הקלוריות היומית המומלצת לפי סוג העבודה והמאמץ:
החישוב נעשה כדלהלן:
לגברים:

$$Val = 66.5 + (13.75 \times \text{משקל}) + (5.003 \times \text{גובה}) - (6.775 \times \text{גיל})$$

לנשים:

$$Val = 655 + (9.563 \times \text{משקל}) + (1.85 \times \text{גובה}) - (4.676 \times \text{גיל})$$

את הערך שהתקבל (Val) יש להכפיל בגורם המאמץ:

1.2 - לאנשים שאינם מתעמלים ועובדים בישיבה.

1.5 - לאנשים שעובדים עבודה פיסית קלה.

2 - לאנשים שעובדים עבודה מאומצת או מתעמלים.

לשם מחשוב ה BMI וצריכת קלוריות מומלצת של אדם, נגדיר מחלקת **Person** בעלת התכונות הבאות:

Id – מחרוזת המייצגת מספר תעודת זהות

gender – תו המייצג מין ('ז' – לזכר, 'נ' – לנקבה)

weight – משקל בק"ג (מספר ממשי)

height – גובה במטרים (מספר ממשי)

age – גיל בשנים (מספר ממשי)

code – תו ('a' / 'b' / 'c') המייצג קוד מאמץ לפי הפירוט הבא:

'a' - עבור אנשים שאינם מתעמלים ועובדים בישיבה.

'b' - עבור אנשים שעובדים עבודה פיסית קלה.

'c' - עבור אנשים שעובדים עבודה מאומצת או מתעמלים.

- א. כתוב עבור המחלקה Person כותרת, תכונות ופעולה בונה המקבלת את ערכי התכונות כפרמטרים.
- ב. כתוב פעולה במחלקה Person, המחזירה את ערך ה-BMI של האדם.
- ג. כתוב פעולה במחלקה Person, המחזירה את צריכת הקלוריות היומית המומלצת, בהתאם לפרטי (תכונות) האדם.

ד. כתוב פעולה חיצונית המקבלת תור Q של אנשים מסוג Person.

- לכל אדם בתור יש להציג את התז שלו, את ערך ה-BMI שלו ואת צריכת הקלוריות המומלצת עבורו.
- מעוניינים לשלוח מכתבי התראה לאנשים בעלי סיכון בריאותי גבוה, לכן עליהם להיות בתחילת התור. יש לעדכן את התור Q כך שבתחילת התור יהיו האנשים ש"קטגוריית המשקל" היא מסוג "תת משקל", ולאחריהם קבוצת האנשים עם "קטגוריית המשקל" היא מסוג "השמנת יתר חמורה", (ראה טבלה 1) (הסדר של האנשים בכל אחת מקטגוריות אלו אינו חשוב). שאר האנשים האחרים יישארו בהתאם לסדר שלהם בתור Q המקורי.

הנח שבמחלקה Person פעולות האחזור והקביעה (set...,get..) לכל תכונה נתונות.

ניתן להשתמש בהן מבלי לממש.

אם אתה נעזר בפעולות פנימיות/חיצוניות נוספות, עליך לממשן.

שאלות שחוברו ע"י דפנה לוי-רשתי

שאלה 1 – שאלת סיפור למבחן מתכונת ב"עיצוב תוכנה"

קופת החולים "אור הבריאה" מחזיקה עבור חברי הקופה מלאי של חיסוני זוסטוואקס כנגד זיהום שלבקת חוגרת. חיסוני זוסטוואקס מתחלקים ל-3 סוגים: חיסון לילד (עד גיל 12), חיסון למבוגר (עד גיל 70) וחיסון לקשיש (מעל גיל זה). על פי הנחיות משרד הבריאות יש להשמיד כל חיסון לאחר 12 חודשים מזמן היצור, ואסור להשתמש בו (אם חיסון יוצר בינואר, יש להשמידו בינואר שנה הבאה).

א. כתוב את חלק התכונות במחלקה "חיסון" (Vaccine).

ב. כתוב פעולה למחלקה חיסון, המקבלת את גילו של החולה ובודקת האם חיסון ספציפי נגד שלבכת חוגרת (zustwaks) מתאים לו.

הקופה מחזיקה את נתוני המלאי של חיסוני zustwaks במערך בגודל 12, כאשר לכל חודש יצור רשומה הכמות הרלבנטית של חיסונים שיוצרו באותו חודש בתא זה: תא 0 במערך עבור חיסונים שיוצרו בחודש ינואר, תא 1 עבור חיסונים שיוצרו בחודש פברואר ועד ל-תא 11 עבור חיסונים שנרכשו בחודש דצמבר. עבור כל חודש רשומים בנפרד הכמויות עבור סוגי החיסונים השונים.

ב. כתוב את חלק התכונות במחלקה "מלאי חיסונים".

ג. כתוב פעולה בונה למחלקה "מלאי חיסונים".

כאשר מתקבלת בקשה מאחד מסניפי הקופה לכמות חיסונים מסוימת, נשלחים תמיד **החיסונים המוקדמים ביותר המותרים לשימוש**.

ד. כתוב פעולה למחלקה "מלאי חיסונים" המקבלת את מספרו הסידורי של חודש הבקשה (1 עבור ינואר וכד'), את סוג החיסון המבוקש ואת הכמות הנדרשת. הפעולה תעדכן את המלאי החדש, על פי קריטריון הקדימות הנ"ל ותחזיר 0 אם כל החיסונים סופקו או את יתרת החיסונים שחסרה במלאי.

ה. כתוב פעולה למחלקה "מלאי חיסונים" המקבלת את מספרו הסידורי של חודש הבקשה, ואת הכמויות הנדרשות עבור סוגי החיסונים השונים. הפעולה תשתמש בפעולה מסעיף ג'. במקרה ולא כל החיסונים סופקו, הפעולה תדאג לפלט שגיאה מתאים.

פתרון לשאלה 1

```
class Vaccine
{
    private string name;
    private int vaccineType;
    private int month;

    public Vaccine(string name, int type, int month)
    {
        this.name = name;
        this.vaccineType = type;
        this.month = month;
    }

    public bool function(int age)
    {
        if (this.name == "zustwaks")
        {
```

```
    if (age < 8)
        return this.vaccineType == 1;
    else
    {
        if (age < 70)
            return this.vaccineType == 2;
        else
            return this.vaccineType == 3;
    }
}
else return false;
}
```

```
class VaccinesStock
{
    private int[,] stock;

    public VaccinesStock()
    {
        this.stock = new int[12,3];
        for (int i = 0; i<12 ; i++){
            for (int j = 0; j < 3; j++)
            {
                stock[i, j] = 0;
            }
        }
    }

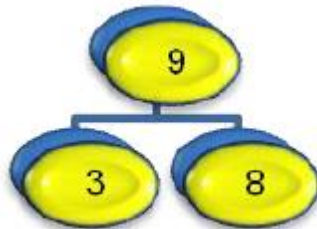
    public int reduceInventory(int month, int type, int num)
    {
        int i = 0;
        int monthIndex = month;
        if (monthIndex == 12) monthIndex = 0;
        while (num > 0 && i < 12)
        {
            int min = Math.Min(num,this.stock[monthIndex, type]);
            if (num == min)
            {
                this.stock[monthIndex, type] -= num;
            }
        }
    }
}
```

```
        num = 0;
    }
    else
    {
        this.stock[monthIndex, type] = 0;
        num -= min;
        monthIndex++;
        if (monthIndex == 12) monthIndex = 0;
        i++;
    }
}
return num;
}

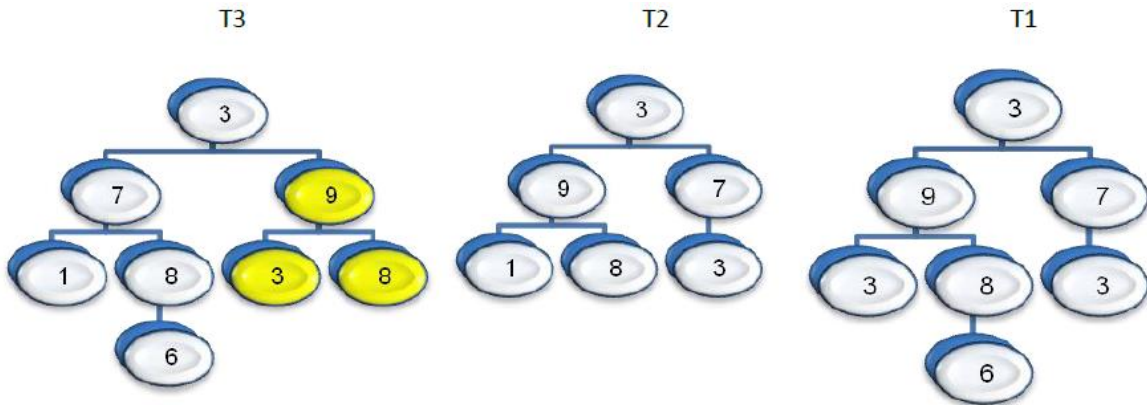
public void reduceInventory(int month, int numType1, int numType2,
                           int numType3)
{
    int num = this.reduceInventory(month, 1, numType1) +
              this.reduceInventory(month, 2, numType2) +
              this.reduceInventory(month, 3, numType3);
    if (num > 0) Console.WriteLine("not all vaccines supplied");
}
}
```

שאלה 2 – שאלת עצים בינאריים למבחן מתכונת ב"עיצוב תוכנה"

א. עץ בינארי Tsmall יקרא עץ בינארי חבוי שלם בעץ בינארי Tlarge אם בעץ Tlarge קיים צומת המהווה שורש של תת עץ שהינו זהה לחלוטין לעץ Tsmall:
לדוגמה, העץ:



הינו עץ בינארי חבוי בעץ T3 בלבד:



כתוב פעולה חיצונית המקבלת שני עצים בינאריים לא ריקים ובודקת האם העץ הראשון חבוי בעץ השני.

ב. עץ בינארי Tsmall יקרא עץ בינארי חבוי חלקי בעץ בינארי Tlarge אם בעץ Tlarge קיים צומת המהווה שורש של תת עץ שתחילתו זהה לחלוטין לעץ Tsmall.

בדוגמה הקודמת, גם בעץ T1 וגם בעץ T3 קיימים עצים בינאריים חבויים חלקית.

כתוב פעולה חיצונית המקבלת שני עצים בינאריים לא ריקים ובודקת כמה פעמים העץ הראשון חבוי חלקי בעץ השני.

```
/**
 * עץ 2 חבוי מלא בעץ 1
 */
static bool IsFullyHidden(BinTreeNode<int> t1, BinTreeNode<int> t2)
{
    return
        ((t1 != null) &&
         (IsEqual(t1, t2) ||
          IsFullyHidden(t1.GetLeft(), t2) ||
          IsFullyHidden(t1.GetRight(), t2)));
}

static bool IsEqual(BinTreeNode<int> t1, BinTreeNode<int> t2)
{
    return
        ( (t1 == t2) || // ריקים שניהם
          ( t1 != null && t2 != null &&
            t1.GetInfo() == t2.GetInfo() &&
            IsEqual(t1.GetLeft(),t2.GetLeft()) &&
            IsEqual(t1.GetRight(),t2.GetRight())));
}
```

```
/******  
* כמה פעמים חבוי חלקית עץ 2 בעץ 1  
* *****/  
  
static int HiddenCount(BinTreeNode<int> t1, BinTreeNode<int> t2)  
{  
    int count = 0;  
    if (t1 == null)  
        return count;  
    count = HiddenCount(t1.GetLeft(),t2) + HiddenCount(t1.GetRight(),t2);  
    if (IsTopEqual(t1, t2)) count++;  
    return count;  
}  
  
static bool IsTopEqual(BinTreeNode<int> t1, BinTreeNode<int> t2)  
{  
    return  
        t2 == null ||  
        ( t1 != null &&  
          t1.GetInfo() == t2.GetInfo() &&  
          IsTopEqual(t1.GetLeft(), t2.GetLeft()) &&  
          IsTopEqual(t1.GetRight(), t2.GetRight()));  
}
```

שאלה 3 – שאלת עצים בינאריים למבחן מתכונת ב"עיצוב תוכנה"

א. תת עץ כביר הוא תת עץ בעץ בינארי ששורשו אינו קטן מערכי כל צאצאיו.
כתוב פעולה חיצונית המקבלת עץ בינארי לא ריק ומחזירה את מספר הצמתים הרב ביותר שיש בתת עץ כביר בעץ.

```
/******  
* מספר הצמתים של תת העץ המקסימלי ששורשו  
* שווה או גדול מכל צאצאיו  
* *****/  
  
static int GreatSubTreeRoot(BinTreeNode<int> t)  
{  
    if (t == null)  
        return 0;  
    if (IsRootGreat(t, t.GetInfo()))  
        return NumOfSuns(t);  
    return Math.Max(GreatSubTreeRoot(t.GetLeft()),  
                    GreatSubTreeRoot(t.GetRight()));  
}
```

```
static bool IsRootGreat(BinTreeNode<int> t, int root)
{
    return t == null ||
        (t.GetInfo() <= root &&
         IsRootGreat(t.GetLeft(), root) && IsRootGreat(t.GetRight(), root));
}

static int NumOfSuns(BinTreeNode<int> bt)
{
    if (bt == null || IfLeave(bt)) return 0;
    if (bt.GetLeft() != null && bt.GetRight() != null)
        return 2 + NumOfSuns(bt.GetRight()) + NumOfSuns(bt.GetLeft());
    else return 1 + NumOfSuns(bt.GetLeft()) + NumOfSuns(bt.GetRight());
}
```

ב. כתוב פעולה חיצונית המקבלת עץ בינארי לא ריק והופכת את שורשו לתת עץ כביר, על ידי שינוי ערכו למספר הקטן ביותר האפשרי.

שאלות שחברו ע"י רחלי צרניחוב

שאלה 1 – שאלת סיפור למבחן מתכונת ב"עיצוב תוכנה"

מתאימה לעיצוב תוכנה על פי תוכנית לימודים הנוכחית.

"בית הכושר" הוא מכון כושר אשר חברות בו משפחות.

עבור כל משפחה (עצם) שמורים במחלקה Member הנתונים הבאים:

- קוד המשפחה
- שם המשפחה
- מספר המבוגרים מעל גיל 18
- מספר הילדים מתחת לגיל 18
- פרטי התא לשמירת חפצים השייך למשפחה.

לכל משפחה במכון יש תא לשמירת חפצים. לתא יש מספר סידורי, קוד תווי המתאר את האזור בו ממוקם התא, ומחיר חדשי הנקבע בהתאם לטיב וגודל התא.

כותרת המחלקה Ta תראה כך:

```
public class Ta
{
    private int code;
    private char area;
    private double price;
```

- א. כתוב את כותרת ותכונות המחלקה Member. (הנח כי המחלקה Ta קיימת)
- ב. הוסף למחלקה Member פעולה אשר מקבלת קוד של משפחה, ומחשבת כמה צריכה המשפחה לשלם למכון בחודש.
התשלום מתבצע באופן הבא: עבור כל מבוגר מעל גיל 18 משלמים 250 ש"ח. עבור ילד 150 ש"ח.
אם יש למשפחה יותר מ 3 ילדים משלמים רק עבור 3 הילדים הראשונים.
יש להוסיף לתשלום החודשי את מחיר התא.
- ג. כתוב ב Main קטע תכנית אשר עובר על פרטי כל המשפחות ומציג עבור כל משפחה את שם המשפחה ואת הסכום לתשלום.
כמו כן, עבור בעלי תא באזור 'a' יש להדפיס את ההודעה הבאה:
"החודש ייעשה שיפוץ בתא שלכם – נא לרוקן אותו".

אם יש צורך בפעולות נוספות של המחלקה Member צריך לכתוב אותן.

שאלה 2 – שאלה אלגוריתמית למבחן מתכונת ב"עיצוב תוכנה"

כתבו פעולה במחלקה הראשית, המקבלת מערך של 10 חנויות ספרים ורשימה של ספרי לימוד הנחוצים לתלמיד. החנויות מסודרות לפי מרחק מבית התלמיד, מהקרובה לרחוקה.
יש להחזיר את מספר חנויות הספרים הקרובות ביותר, בהן על התלמיד לבקר כדי לרכוש את כל הספרים הנחוצים לו.

תכונות של עצם מטיפוס חנות ספרים

shopName	שם החנות
add	כתובת (עצם מטיפוס כתובת מכיל רחוב מספר ויישוב)
bookList	רשימת ספרים המצויים בחנות

תכונות של עצם מטיפוס ספר לימוד

bookName	שם הספר
author	שם המחבר
price	מחיר

- ניתן להניח שקיימות פעולות בונות, מאחזרות ומעדכנות למחלקה הנ"ל. (אין צורך לכתוב אותן!)
- אם ברצונכם להוסיף פעולות עזר למחלקות עליכם לכתוב אותן.

תכנים נדרשים לפתרון המשימה:

- מערך של עצמים מורכבים שאחת מהתכונות שלו היא רשימה.
- רשימה.
- סריקת רשימה וחיפוש איבר ברשימה שהיא תכונה של עצם במערך.

שאלות שחוברו ע"י דני קשת

שאלה 1 – שאלת סיפור למבחן מתכונת ב"עיצוב תוכנה"

נגדיר את 3 המחלקות הבאות – מילה משפט ופיסקה.
התכונות של מחלקת "מילה" הן 2 מחרוזות – המילה עצמה והמילה הארוכה ביותר שנוצרה (תכונה סטטית, כמובן)

התכונה של משפט היא חוליה ראשונה בשרשרת חוליות גנריות מסוג "מילה"
התכונות של מחלקת "פיסקה" הן מערך של עצמים מסוג "משפט" ומספר המציין את מספר המשפטים.
גודל המערך הוא קבוע - 10, ובהתחלה הוא כמובן ריק.

במחלקת הבדיקה, ב- MAIN, נקלוט פיסקה (בצורה של מחרוזת אחת ארוכה) ונבנה עצם ממחלקת פיסקה כפי שהגדרנו.

כמובן שעלינו לפרק את פיסקת הקלט למשפטים ומילים .

משפט מסתיים כשיש נקודה לאחריו.

מילה מסתיימת כשמגיע רווח.

בשאלה זו, הנח שאין סימנים אחרים כמו פסיק וסימני קריאה ושאלה.

טיפ לסיוע – בשפת C# קיימת פעולה בשם Split הפועלת על מחרוזת Split(char delimiter) הפעולה מחזירה מערך של מחרוזות לפי הדלימיטר (המפריד)

א. ממש את הפעולה הבונה של Word.

ב. ממש את הפעולה AddWord(Node<Word> nw). כתוב באיזה מחלקה מימשת פעולה זו.

ג. ממש את פעולת ToString() של מחלקת Piska. הפעולה תדפיס על המסך את כל הפיסקה (ולמעשה תשכפל את כל מה שקלטנו).

ד. בתכנית הבדיקה, אחרי שבנית את עצם p1 מטיפוס Piska, הדפס את המילה האחרונה במשפט השני. ניתן להניח שיש את כל פעולות ה-Get הנדרשות

ה. הדפס את המילה הארוכה ביותר בפיסקה (או אחת מהן) – רמז: אם קראת נכון את השאלה, זה הסעיף הקל ביותר.

שאלה 2 – שאלת מעקב למבחן מתכונת ב"עיצוב תוכנה"

א. לפניך מספר מחלקות. עקוב אחרי הפעולה Main שבמחלקת Program ורשום את הפלט.

במעקב עליך לבנות מפת עצמים והפניות עם ערכי התכונות של כל עצם

```
public class Point
{
    private double x, y;
    public Point(double x, double y)
    {
```

```
        this.x = x;
        this.y = y;
    }
    public double GetX()
    {
        return this.x;
    }
    public double GetY()
    {
        return this.y;
    }
    public override string ToString()
    {
        return this.x + "," + this.y; ;
    }
    public int F()
    {
        return 100;
    }
}

public class TheClass1
{
    private Point[] ps;
    public TheClass1(int n)
    { //Random rn=new Random();
        ps = new Point[n];
        for (int i = 0; i < n; i++)
        {
            ps[i] = new Point(i,1+(i*17)%10);
        }
    }
    public string F()
    {
        return "i'm F of TheClass1";
    }
    public Point[] GetPoints()
    {
        return this.ps;
    }
}

public class Linear
{
```

```
private double a, b;
public Linear(double a, double b)
{
    this.a = a;
    this.b = b;
}
public bool OnIt(Point p)
{
    if (p.GetY() == this.a * p.GetX() + this.b)
        return true;
    return false;
}
}

public class Od
{
    private TheClass1 tc1;
    private List<Point> lst;
    private Linear lin;
    public Od(int n, double s, double h)
    {
        tc1 = new TheClass1(n);
        lst = new List<Point>();
        lin = new Linear(s, h);
    }
    public string F()
    {
        return "i'm from Od " + tc1.F();
    }
    public int Do()
    {
        Point[] pts=tc1.GetPoints();
        int cnt = 0;
        for (int i = 0; i < pts.Length; i++)
        {
            if (this.lin.OnIt(pts[i]))
            {
                lst.Insert(null, pts[i]);
                Console.WriteLine(lst.GetFirst()); ;
                cnt++;
            }
        }
        return cnt;
    }
}
```

```
}  
class Program  
{  
    static void Main(string[] args)  
    {  
        Od more = new Od(5, 2, 1);  
        Console.WriteLine(more.F());  
        Console.WriteLine(more.Do());  
        Console.ReadLine();  
    }  
}
```

ב. יוסי טוען – מכיוון שיש 3 פעולות בשם F(), יש כאן מקרה של העמסת פעולות

(Method overloading)

אבי טוען – אתה צודק. אבל שים לב שבמחלקת Point הפעולה מחזירה int ובמחלקות האחרות מוחזר string, לכן רק במחלקות האחרות יש העמסת פעולות.

ראובן טוען – אם במחלקת Od היתה הפעולה שלנו F עם פרמטרים אז היה העמסת פעולות. משה טוען – מצטער חברים- שלושתכם טועים...

הסבר – מי מהתלמידים צודק ומה הטעות של כל אחד מהאחרים?

ג. בונים (5 נק') – במשפט אחד בלבד – מה מבצעת התכנית? (התעלם מההדפסה הראשונה)