

מדעי המחשב ב'
פתרון בחינת הבגרות - בשפות Java, C#
פרק א - עיצוב תכנה

שאלה 1:

פתרון בשפת C#:

נכתב ע"י ראמי ג'באלי

```
// פעולה המקבלת רשימה lst לא ריקה של מספרים, ומחזירה רשימת הסכומים העולים.  
public static List<int> SortSum(List<int> list)  
{  
    Node<int> pos = list.GetFirst();  
    List<int> listSum = new List<int>();  
    Node<int> pos2 = listSum.GetFirst();  
    int sum = 0;  
    while (pos.GetNext() != null)  
    {  
        if (pos.GetInfo() < pos.GetNext().GetInfo())  
            sum = sum + pos.GetInfo();  
        else  
        {  
            pos2 = listSum.Insert(pos2, sum + pos.GetInfo());  
            sum = 0;  
        }  
        pos = pos.GetNext();  
    }  
    pos2 = listSum.Insert(pos2, sum + pos.GetInfo());  
    return listSum;  
}
```

פתרון בשפת Java:

```
// פעולה המקבלת רשימה lst לא ריקה של מספרים, ומחזירה רשימת הסכומים העולים.
public static List<Integer> sortSum(List<Integer> list)
{
    Node<Integer> pos = list.getFirst();

    List<Integer> listSum = new List<Integer>();
    Node<Integer> pos1 = listSum.getFirst();

    int x, y, sum;
    x = pos.getInfo();
    sum = x;
    pos = pos.getNext();
    while (pos != null)
    {
        y = pos.getInfo();
        if (y > x)
            sum = sum + y;
        else
        {
            pos1 = listSum.insert(pos1, sum);
            sum = y;
        }
        x = y;
        pos = pos.getNext();
    }
    pos1 = listSum.insert(pos1, sum);
    return listSum;
}
```

שאלה 2:

סימטרי-סכומים (A, i, j, val)

- } פעולה המקבלת מערך A בגודל N, מקומות i, j, ו- val = הסכום המבוקש. הפעולה מחזירה "אמת" אם לכל זוג איברים הנמצאים במרחק סימטרי ממרכז המערך סכום זהה, ו-"שקר" אחרת. הנחות: N זוגי, i, j מקומות במערך כך ש- i אינו מקום ראשון, ו- j אינו מקום אחרון
- (1) אם $i > j$ אזי - החזר "אמת"
- (2) אם $A[i] + A[j] \neq val$ אזי - החזר "שקר"
- (3) החזר - סימטרי-סכומים (A, i+1, j-1, val)

נכתב ע"י ראמי ג'באלי

פתרון בשפת C#:

```
using System;
class T2_2003
{
    // פעולה קולטת מערך
    public static void Read(int[] a)
    {
        for (int i = 0; i < a.Length; i++)
            a[i] = int.Parse(Console.ReadLine());
    }
    // פעולה מחזירה אמת אם הערך הוא סימטרי אחרת תחזיר הפעולה שקר
    public static bool IsSyemtric(int[] a, int i, int j, int x)
    {
        if (i > j)
            return true;
        else if (a[i] + a[j] != x)
            return false;
        else return IsSyemtric(a, i + 1, j - 1, x);
    }
    // תוכנית ראשית
    static void Main(string[] args)
    {
        int[] a = new int[6];
        Read(a);
        int x = a[0] + a[a.Length - 1];
        if (IsSyemtric(a, 0, a.Length - 1, x))
            Console.WriteLine("Yes");
        else Console.WriteLine("No");
    }
}
```

פתרון Java :

```
public class T2_2003
{
    /**
     * תרגיל 2 - בגרות 2003 - סימטרי סכומים
     */
    public static void main(String[] args)
    {
        int [] arr = {7, 3, 5, 4, 6, 2};
        int N = arr.length;

        int val = arr[0] + arr[N - 1] ;
        if (symmetricSum (arr, 1, N-2, val) )
            System.out.println("כן") ;
        else
            System.out.println("לא") ;
    }

    //--- פעולה המקבלת מערך של שלמים, שני אינדקסים במערך וערך
    //--- ומחזירה "אמת" אם סכום שני התאים במקומות הנתונים שווים
    //--- לערך הנתון, ו-"שקר" אחרת.
    //--- הנחה: המקומות הם אינדקסים חוקיים במערך
    static boolean symmetricSum (int [] arr, int i, int j, int val)
    {
        if (i>j)
            return true ;
        if (arr[i] + arr[j] != val)
            return false ;
        return symmetricSum (arr, i+1 , j-1, val) ;
    }
}
```

ניתן למלא את המערך בפעולה:
arrKelet (arr) הקולטת ערכים למערך.

לסוף 3:

האלגוריתם מתורגם לשפת C#.

```
static List<int> Sod1 (BinTreeNode <int> t)
{
    List<int> lst = new List<int> ();
    lst = Sod2 (t, lst);
    return lst;
}

static List<int> Sod2 (BinTreeNode <int> t, List<int> lst)
{
    if (t != null)
    {
        int x = t.GetInfo();
        if (Sod3(t.GetLeft() && Sod3(t.GetRight()))
            lst.Insert (null, x);
        lst = sod2(t.GetLeft(), lst);
        lst = sod2 (t.GetRight(), lst);
        return lst;
    }

    static bool sod3 (BinTreeNode <int> t, int x)
    {
        if (t == null) return true;
        int a = t.GetInfo();
        if (x == a) return false;
        return sod3(t.GetLeft(), x) && sod3(t.GetRight(), x);
    }
}
```

האלגוריתם מתורגם לשפת Java.

```

static List<Integer> sod1 (BinTreeNode <Integer> t)
{
    List<Integer> lst = new List<Integer> ();
    lst = sod2 (t, lst);
    return lst;
}

static List<Integer> sod2 (BinTreeNode <Integer> t, List<Integer> lst)
{
    if (t != null)
    {
        int x = t.getInfo();
        if (sod3(t.getLeft() && sod3(t.getRight()))
            lst.insert (null, x);
        lst = sod2(t.getLeft(), lst);
        lst = sod2 (t.getRight(), lst);
        return lst;
    }

    static boolean sod3 (BinTreeNode <Integer> t, int x)
    {
        if (t == null) return true;
        int a = t.getInfo();
        if (x == a) return false;
        return sod3(t.getLeft(), x) && sod3(t.getRight(), x);
    }
}

```

- א. סוד-3 - מחזיר "אמת" אם x אינו מופיע בעץ T, ו"שקר" אחרת.
- ב. סוד-1 - מחזיר רשימה שבה כל ערכי x שאינם מופיעים בתתי העצים ש-x נמצא בשורש שלהם. (אם לצומת המכיל x יש צאצא השווה לו, x לא יהיה ברשימה (הסריקה נעשית בסדר תוכי). שים ♥: יתכן ו- x יופיע כאח של שורש תת-העץ או כצאצא של האח. במקרה זה x יופיע ברשימה.
- ג. אם כל האיברים ב-T שונים זה מזה, יופיעו כל הצמתים ברשימה.

שאלה 4:

א. ייצוג המידע:

לקוח-בודד = אובייקט מסוג Person

קבוצה = מערך בגודל 2 עד 8 של אובייקטים מסוג Person

(הפעולה הבונה תקבל את גודל הקבוצה ותיצור מערך בגודל זה. יש לבדוק שגודל הקבוצה בין 2 ו-8)

סירה = מערך בגודל 8 של אובייקטים מסוג Person.

class Single
String name;
Single ();
הפעולה get הפעולה toString()

class Group
int num; // מספר בין 2-8
Person [] arr;
Group (int num);
פעולות get ו-set הפעולה toString()

class Boat
static final int num = 8;
Person [] arr;
Boat ();
פעולות get ו-set הפעולה toString() פעולות הוספה לסירה : הוספת בודד הוספת קבוצה

מתקן-מים-גועשים :

סירות = מערך בגודל 14 שכל איבר מטיפוס סירה.

תור-בודדים = תור שכל איבר בו מטיפוס לקוח-בודד.

תור-קבוצות = תור שכל איבר בו מטיפוס קבוצה.

class StormyWater
static final N = 14; // מס' הסירות במערך
Boat [] boat;
Queue <Single> singles;
Queue <Group> groups;
StormyWater();
כל הפעולות המופיעות בממשק שלהלן

ב. ממשק עברי למחלקה StormyWater :

פעולה בונה למתקן, המחזירה מתקן שבו מערך הסירות מאותחל, תור הבודדים ותור הקבוצות מאותחלים וריקים.	StormyWater()
פעולה המאתחלת סירה במקום i במערך הסירות. שים ♥ : פעולה זו מפעילה את הפעולה הבונה של המחלקה Boat	אתחל-סירה (i)
פעולה המקבלת קבוצה ומכניסה אותה לתור-הקבוצות במתקן. הנחה : הקבוצה תקינה.	הכנס-לתור-קבוצות (קבוצה)
פעולה המוסיפה לקוח-בודד לתור-בודדים.	הכנס-לתור-בודדים ()
פעולה המוציאה ומחזירה קבוצה מתור-הקבוצות. הנחה : תור הקבוצות אינו ריק.	הוצא-מתור-קבוצות ()
פעולה המוציאה ומחזירה לקוח-בודד מתור-הבודדים. הנחה : תור הבודדים אינו ריק.	הוצא-מתור-בודדים ()
פעולה מחזירה את מספר האנשים בקבוצה הנמצאת בראש תור-הקבוצות. הנחה : תור הקבוצות אינו ריק.	אחזר-גודל-קבוצה ()
פעולה המחזירה "אמת" אם תור-הקבוצות ריק, ו"שקר" אחרת.	תור-קבוצות-ריק? ()
פעולה המחזירה "אמת" אם תור-הבודדים ריק, ו"שקר" אחרת.	תור-בודדים-ריק? ()

הפעולות שלהלן הינן חלק מהמחלקה Boat :

פעולה המחזירה את מספר המקומות הפנויים בסירה הנוכחית.	מקומות-פנויים-בסירה ()
פעולה המשבצת אדם בסירה הנוכחית, בתנאי שיש בה מקום פנוי.	שבץ-בודד-בסירה ()
פעולה המשבצת קבוצה בסירה הנוכחית, בתנאי שיש בה מקום פנוי לגודל הקבוצה. הנחה : גודל הקבוצה בין 2 ו-8.	שבץ-קבוצה-בסירה (קבוצה)
פעולה החזירה את הסירה הנוכחית ריקה מאנשים.	רוקן-סירה ()