

## פרק 13 – פתרון בעיות

### הצגת הפרק

#### מטרת הפרק

פרק זה נועד להכין את התלמיד לפתרון בעיות המשלבות פיתוח אלגוריתמי יחד עם הגדרת מחלקות מתאימות לפתרון הבעיה, כדוגמת בעיות חלק ג' של בחינת הבגרות. בפרק זה בבגרות על הנבחן לנתח בעיה אלגוריתמית ולממש אותה באמצעות מחלקה מתאימה.

### דגשים ודידקטיקה

- הפרק מחולק למספר קטגוריות, כאשר כל קטגוריה מציגה בעיות בעלות פן שונה. בסוף כל קטגוריה מוצגות מספר בעיות באותה הרוח. בסוף הפרק מוצגות שאלות נוספות מכל הקטגוריות ללא סיווג כלשהו. מומלץ לעבור על הקטגוריות השונות, להציג לתלמידים את סוגי הבעיות השונות, ולבסוף לתרגל את הבעיות הנוספות באופן שהם יצטרכו למצוא מהו הפתרון האלגוריתמי המתאים לפתרון הבעיה.

## פתרונות

**שאלה 13.2** בהשראת בגרות קיץ 2005 תשס"ה, שאלה 9

נגדיר "פרח" במערך באופן הבא: הפרח מורכב מ-5 איברים מתוך תת-מערך בגודל  $3 \times 3$ . האיבר המרכזי בתת-מערך הוא "לב הפרח". ארבעת האיברים הצמודים לו בפינותיו הם "עלי הכותרת" של ה"פרח". הערך של "לב הפרח" שווה לסכום ערכי "עלי הכותרת" של ה"פרח" ובכל "פרח" חייבים להיות בדיוק 4 "עלי כותרת".

דוגמה: במערך בגודל  $5 \times 4$  שלפניכם יש "פרח" אחד:

0	0	3	3
2	0	2	1
0	0	2	3
1	4	8	11
0	-2	9	7

"עלה כותרת" ←

← "לב הפרח"

א. כתבו מחלקת "שדה פרחים" המכילה מערך דו-מימדי המייצג שדה. הפעולה הבונה מקבלת מערך-דו מימדי, ומאתחלת את השדה.

ב. כתבו פעולה המקבלת שני מספרים שלמים המציינים מקום של איבר במערך (אינדקסים), המספר הראשון מציינ שורה, והמספר השני מציינ עמודה. הפעולה תחזיר `true` אם איבר זה הוא "לב הפרח" של "פרח" במערך; אחרת – הפעולה תחזיר `false`.

ג. מערך יקרא "פרחוני" אם יש בו לפחות 5 "פרחים". כתבו פעולה שתבדוק האם המערך הוא "פרחוני", ותחזיר ערך בהתאם. השתמשו בפעולה שכתבתם בסעיף ב'. שימו לב, הפרחים יכולים להיות חופפים זה לזה, כלומר מותר לפרח אחד להכיל לב או עלי כותרת השייכים לפרח אחר.

### תשובה 13.2

```
/* מחלקה לייצוג שדה פרחים */
public class FlowersField
{
    int[,] field;
    public FlowersField(int[,] field)
    {
        this.field = field;
    }
    public bool IsFlower(int i, int j)
    {
        if (i<=0 || j<=0)
            return false;
        if (i>=field.GetLength(0)-1 || j>=field.GetLength(1)-1)
            return false;
        int sum = field[i-1,j-1]+field[i+1,j-1]+
                field[i-1,j+1]+field[i+1,j+1];
        return (field[i,j] == sum);
    }
}
```

```

}
public bool IsFlowery()
{
    int count = 0;
    for (int i = 1; i < field.GetLength(0)-1; i++)
        for (int j = 1; j < field.GetLength(1)-1; j++)
            if (IsFlower(i,j))
                count++;
    return count>=5;
}
}

```

### 13.3 שאלה

באולימפיאדת הארנבים מתקיימת תחרות בשלוש קטגוריות שונות: ריצה לגזר, ריצה לכרוב וריצה לברוקולי. באולימפיאדה מחולקות שתי מדליות זהב. את הראשונה מקבל הארנב שהתוצאה הממוצעת שלו, המורכבת משלושת התחרויות היא הטובה ביותר. את השנייה מקבל הארנב שזמן הריצה שלו הוא הטוב ביותר מבין כל התוצאות של כל הארנבים בכל הקטגוריות. פתחו וממשו אלגוריתם העוזר למארגני התחרות לבחור את המנצחים. האלגוריתם יקלוט בתחילה את מספר הארנבים המתחרים, לאחר מכן יקלוט רשימה של שמות הארנבים ותוצאות הביצוע של כל ארנב בכל קטגוריה. פלט האלגוריתם יהיה שמות שני מקבלי מדליות זהב.

**הדרכה:** בדומה לבעיה 2, גם בבעיה זו יש לשמור אוסף, כלומר, הפעולה הראשית תכיל מערך של עצמים. בבעיה זו כל עצם הוא ממחלקת "ארנב". תכונות ה"ארנב הם: שם הארנב ומערך המכיל את שלושת התוצאות שלו. הפעולות על עצם מסוג ארנב יאפשרו לקבל את ממוצע התוצאות של הארנב וכן את תוצאת הריצה הטובה ביותר שלו. הפעולה הראשית תקלוט את נתוני הארנבים תייצר אותם ותשמור אותם במערך, תחפש את הארנב בעל התוצאה הממוצעת הטובה ביותר ואת הארנב בעל התוצאה הגבוהה ביותר, ותציג את שמם.

### 13.3 תשובה

```

/* מחלקת ארנב אולימפי */
class RabbitOlympics
{
    private string name;
    private double[] times;
    public RabbitOlympics(string name, double[] times)
    {
        this.name = name;
        this.times = new double[times.Length];
        for (int i = 0; i < times.Length; i++)
            this.times[i] = times[i];
    }
    public double GetBestResult()
    {
        double min=this.times[0];
        for (int i=1; i<times.Length; i++)
            if (this.times[i] < min)
                min = this.times[i];
        return min;
    }
}

```

```

    public double GetAverageResult()
    {
        double sum = 0;
        int i;
        for (i = 0; i < this.times.Length; i++)
            sum += this.times[i];
        return (sum/i);
    }
    public string GetName()
    {
        return name;
    }
} //RabbitOlympics

/* המחלקה הראשית */
using System;
public class Rabbits
{
    public static void Main()
    {
        // הצהרה על משתנים
        int numOfRabbits;
        RabbitOlympics[] rabbits;
        string rabbitStr;
        const int NUM_RACES = 4;
        double[] rabbitScoresArray;
        Console.WriteLine("Insert number of rabbits: ");
        numOfRabbits = int.Parse(Console.ReadLine());
        rabbits = new RabbitOlympics[numOfRabbits];
        rabbitScoresArray = new double[NUM_RACES];
        // קלט
        for (int i = 0; i < numOfRabbits; i++)
        {
            Console.WriteLine("Insert a rabbit name: ");
            rabbitStr = Console.ReadLine();
            for (int scoreIndex = 0; scoreIndex < NUM_RACES;
                scoreIndex++)
            {
                Console.WriteLine("Insert the rabbit score: ");
                rabbitScoresArray[scoreIndex] =
                    int.Parse(Console.ReadLine());
            }
            rabbits[i] = new RabbitOlympics(rabbitStr,
                rabbitScoresArray);
        }

        RabbitOlympics bestGold = rabbits[0];
        RabbitOlympics averageGold = rabbits[0];

        for (int i=1; i<rabbits.Length; i++)
        {
            if (rabbits[i].GetBestResult() <

```

```

        bestGold.GetBestResult())
        bestGold = rabbits[i];
        if (rabbits[i].GetAverageResult() <
            averageGold.GetAverageResult())
            averageGold = rabbits[i];
    }
    Console.WriteLine("Best rabbit is : {0}",
        bestGold.GetName());
    Console.WriteLine("Average best rabbit is {0}",
        averageGold.GetName());
} //Main
} // Rabbits

```

### שאלה 13.5

נצחיה, המורה להיסטוריה, מעוניינת לדעת את ממוצע כל אחת מ-6 הכיתות שלה במבחן המשווה, וכן את הממוצע השכבה. צחי שוב ניסה את מזלו בהכנת קטע קוד לפתרון הבעיה:

```

const int NUM_OF_CLASSES = 6;
int classSize;
int classSum = 0;
int totalSum = 0;
int totalNumOfStudents = 0;
int grade;
for (int i = 0; i < NUM_OF_CLASSES; i++)
{
    Console.Write("Enter number of students in class: {0}", (i+1));
    classSize = int.Parse(Console.ReadLine());
    totalNumOfStudents = totalNumOfStudents + classSize;
    for (int j = 0; j < classSize; j++)
    {
        grade = int.Parse(Console.ReadLine());
        classSum = classSum + grade;
    }
    Console.WriteLine( (double)classSum / classSize );
    totalSum = totalSum + classSum;
}
Console.WriteLine( (double)totalSum / totalNumOfStudents );

```

- א. האם קטע התוכנית משיג את מטרתו?
- ב. אם קטע התוכנית שגוי, הציעו תיקון כך שקטע התוכנית ישיג את מטרתו עבור כל קלט אפשרי.
- ג. אילו תבניות שולבו בקטע התוכנית?

## תשובה 13.5

- א. קטע התוכנית אינו משיג את מטרתו. צובר הציונים הכיתתי אינו מאותחל ל-0 לפני תחילת קליטת הציונים של כיתה חדשה.
- ב. קטע התוכנית המתוקן:

```
const int NUM_OF_CLASSES = 6;
int classSize;
int totalSum = 0;
int totalNumOfStudents = 0;
int grade;
for (int i = 0; i < NUM_OF_CLASSES; i++)
{
    Console.WriteLine("Enter number of students in class: {0}", (i+1));
    classSize = int.Parse(Console.ReadLine());
    totalNumOfStudents = totalNumOfStudents + classSize;
    int classSum = 0;
    for (int j = 0; j < classSize; j++)
    {
        grade = int.Parse(Console.ReadLine());
        classSum = classSum + grade;
    }
    Console.WriteLine( (double)classSum / classSize );
    totalSum = totalSum + classSum;
}
Console.WriteLine( (double)totalSum / totalNumOfStudents );
```

- ג. שתי התבניות ששולבו הן ממוצע של סדרת מספרים.

---

## שאלה 13.8

מנהלי קבוצת הכדורסל "גי'רפות בע"מ" מעוניינים לקבל מידע על ביצועי עשרת השחקנים בקבוצה. בעת משחק, ברגע ששחקן קולע לסל, מקבלים כקלט את מספר השחקן (מספר בין 1 ל-10) ואת מספר הנקודות שקלע (מספר בין 1 ל-3). בסיום משחק מתקבל כקלט המספר -1 כמספר שחקן כדי לסיים את הקליטה. בסיום המשחק מנהלי הקבוצה מעוניינים לדעת: מה מספר השחקן שקלע את מספר הסלים המרבי? מה מספר השחקן שצבר את מספר הנקודות המרבי? מה מספר השחקן שקלע את מספר הסלים הנמוך ביותר? כתוב תוכנית בשפת Java אשר תספק למנהלי הקבוצה את המידע הדרוש. עליכם לבנות מחלקה המגדירה משחק כדורסל. עצם מסוג זה ירכז את נתוני השחקנים במשחק ויספק את המידע הדרוש. אם קיימת יותר מתשובה אחת (למשל, שני שחקנים צברו את מספר הנקודות המרבי), יוחזר מספר השחקן הוותיק יותר (שמספרו נמוך יותר).

**הדרכה:** הגדירו לעצם מסוג "משחק כדורסל" פעולה המקבלת נתונים של קליעה אחת: מספר שחקן הקולע ומספר הנקודות שקיבל. פעולה זו תעדכן את התכונות המתאימות אשר ישמרו את המידע הדרוש למנהלי הקבוצה.

```

/* מחלקת משחק כדורסל */
public class BasketballGame
{
    private int[] points;
    private int[] baskets;
    const int NUM_OF_PLAYERS = 10;
    public BasketballGame()
    {
        points = new int[NUM_OF_PLAYERS];
        baskets = new int[NUM_OF_PLAYERS];
        for (int i = 0; i < NUM_OF_PLAYERS; i++)
        {
            points[i] = 0;
            baskets[i] = 0;
        }
    }
    public void AddPoints(int player, int count)
    {
        points[player-1] += count;
        baskets[player-1]++;
    }
    public int GetMaxBasketsPlayer()
    {
        int maxPlayer=0;
        for (int i=1; i<NUM_OF_PLAYERS; i++)
            if (baskets[i] > baskets[maxPlayer])
                maxPlayer=i;
        return maxPlayer+1;
    }
    public int GetMaxPointPlayer()
    {
        int maxPlayer = 0;
        for (int i = 1; i < NUM_OF_PLAYERS; i++)
            if (points[i] > points[maxPlayer])
                maxPlayer=i;
        return maxPlayer+1;
    }
    public int GetMinBasketsPlayer()
    {
        int minPlayer=0;
        for (int i = 1; i < NUM_OF_PLAYERS; i++)
            if (baskets[i] < baskets[minPlayer])
                minPlayer = i;
        return minPlayer+1;
    }
}

/* המחלקה הראשית המשתמשת במחלקת משחק כדורסל */
using System;
public class UseBasketBallGame
{

```

```

public static void Main()
{
    BasketballGame game = new BasketballGame();
    Console.WriteLine("Insert player number (1-10): ");
    int player = int.Parse(Console.ReadLine());
    while (player != -1)
    {
        Console.WriteLine("Insert number of points (1-3): ");
        int points = int.Parse(Console.ReadLine());
        game.AddPoints(player, points);
        Console.WriteLine("Insert player number (1-10): ");
        player = int.Parse(Console.ReadLine());
    }
    Console.WriteLine("Player with max number of baskets: " +
        game.GetMaxBasketsPlayer());
    Console.WriteLine("Player with max number of points: " +
        game.GetMaxPointPlayer());
    Console.WriteLine("Player with min number of baskets: " +
        game.GetMinBasketsPlayer());
}
}

```

### שאלה 13.10

א. נתונות שתי סדרות של מספרים. עליכם לפתח אלגוריתם היוצר סדרה שלישית, המכילה את המספרים הקיימים בשתי הסדרות. הגדירו מחלקה בשם MatchNumbers, במחלקה יוגדרו שני מערכים כתכונות ותוגדר פעולה שמחפשת את המספרים הקיימים בשני המערכים ומחזירה מערך חדש המכיל מספרים אלו.

ב. כעת נתון כי שני המערכים ממוינים בסדר עולה. האם האלגוריתם ישתנה בעקבות מידע זה? כתבו מחדש את הפעולה לפי האלגוריתם החדש.

### תשובה 13.10

א.

```

using System;
public class MatchNumbers
{
    int[] arr1;
    int[] arr2;
    public MatchNumbers(int[] a1, int[] a2)
    {
        arr1 = a1;
        arr2 = a2;
    }
    public int[] GetMatchNumbers()
    {
        int[] res = new int[Math.Min(arr1.Length, arr2.Length)];
        int ires = 0;
        for (int i1 = 0; i1 < arr1.Length; i1++)
        {
            bool found = false;
            for (int i2 = 0; i2 < arr2.Length && !found; i2++)
                if (arr1[i1]==arr2[i2])

```



```

        found = true;
    if (found)
    {
        res[ires]=arr1[i1];
        ires++;
    }
}

//צירת מערך התשובות בגודל מדויק
int[] ret = new int[ires];
for (int i = 0; i < ret.Length; i++)
    ret[i]=res[i];
return ret;
}
}

```

ב. אם שני המערכים ממוינים בסדר עולה, ניתן לשנות את האלגוריתם כך שיהיה יעיל יותר, באופן הבא:

```

public int[] GetMatchSortedNumbers()
{
    int[] res = new int[Math.Min(arr1.Length, arr2.Length)];
    int ires = 0, i1 = 0, i2 = 0;
    while (i1 < arr1.Length && i2 < arr2.Length)
    {
        if (arr1[i1]==arr2[i2])
        {
            res[ires]=arr1[i1];
            ires++;
            i1++;
            i2++;
        }
        else if (arr1[i1]<arr2[i2])
            i1++;
        else //arr1[i1]>arr2[i2]
            i2++;
    }
    //צירת מערך התשובות בגודל מדויק
    int[] ret = new int[ires];
    for (int i = 0; i < ret.Length; i++)
        ret[i] = res[i];
    return ret;
}
}

```

### שאלה 13.11

כתבו את מחלקת "סכומי ספרות". הגדירו במחלקה תכונה יחידה: `nums` – מערך חד-ממדי של מספרים שלמים, והגדירו שתי פעולות:

א. פעולה המחזירה מערך אשר כל אחד מאיבריו הוא סכום הספרות של האיבר המקביל לו ב-`nums`. לדוגמה עבור המערך `nums` הבא:

123	45	532	12	75
-----	----	-----	----	----

יוחזר המערך:

6	9	10	3	12
---	---	----	---	----

ב. "ספרת הסכום" היא הספרה אשר מתקבלת כאשר סוכמים ספרות מספר שוב ושוב עד אשר מתקבלת ספרה יחידה. הגדירו פעולה המחזירה מערך אשר כל אחד מאיבריו הוא "ספרת הסכום" של האיבר המקביל לו ב-`nums`. לדוגמה, עבור המערך `nums` מסעיף א יוחזר המערך:

6	9	1	3	3
---	---	---	---	---

**הדרכה:** הגדירו במחלקה זו פעולת עזר פרטית המקבלת מספר וסוכמת את ספרותיו.

### תשובה 13.11

```
public class DigitsSum
{
    int[] nums;
    public DigitsSum(int[] arr)
    {
        nums = arr;
    }
    public int[] GetDigitsSum()
    {
        int[] arr = new int[nums.Length];
        for (int i = 0; i < arr.Length; i++)
            arr[i] = GetDigitsSum(nums[i]);
        return arr;
    }
    public int[] GetDigitsSumSum()
    {
        int[] arr = new int[nums.Length];
        for (int i=0; i<arr.Length; i++)
        {
            int sum = GetDigitsSum(nums[i]);
            while (sum >= 10)
                sum = GetDigitsSum(sum);
            arr[i] = sum;
        }
        return arr;
    }
    private int GetDigitsSum(int num)
    {
        int sum = 0;
        while (num>0)
        {
```

```

        sum += num % 10;
        num /= 10;
    }
    return sum;
}
}

```

## שאלות נוספות

### שאלה 1

לפניכם מחלקת שירות להצפנה, לפענוח ולבדיקת סיסמאות. אלגוריתם ההצפנה פועל כך: כל אות אנגלית הופכת להיות האות **הבאה** וכל ספרה הופכת להיות הספרה **הקודמת**. ההצפנה פועלת באופן מעגלי כך שהאות הבאה אחרי האות z היא a והספרק הקודמת לספרה 0 היא הספרה 9. לדוגמה הסיסמה: abz9130 תוצפן כך: bca8029.

המחלקה כוללת את הפעולות הבאות:

◆ **encrypt** – פעולת הצפנה המקבלת סיסמה ומחזירה סיסמה מוצפנת לפי האלגוריתם שתואר לעיל.

◆ **decrypt** – פעולת פיענוח המקבלת סיסמה מוצפנת ומחזירה סיסמה גלויה.

◆ **IsLegal** – פעולת לבדיקת חוקיות סיסמה. הפעולה מקבלת סיסמה (גלויה או מוצפנת) ובודקת את חוקיותה. סיסמה תחשב כחוקית אם היא עונה על הדרישות הבאות: אורכה 6-8 תווים, היא מורכבת מאותיות אנגליות ומספרות בלבד וכוללת לפחות אות אחת וספרה אחת.

השלימו את פעולות המחלקה שלהלן:

```

/* מחלקת שירות לסיסמאות */
public class Password
{
    // הפעולה מקבלת סיסמה ומחזירה סיסמה מוצפנת
    public static string encrypt(string pass)
    {
        ...
    }

    // הפעולה מקבלת סיסמה מוצפנת ומחזירה סיסמה גלויה
    public static string decrypt(string pass)
    {
        ...
    }

    // הפעולה מקבלת סיסמה ומחזירה האם הסיסמה חוקית
    public static bool IsLegal(string pass)
    {
        ...
    }
}

```

```

/* מחלקת שירות לסיסמאות */
public class Password
{
    // הפעולה מקבלת סיסמה ומחזירה סיסמה מוצפנת
    public static string encrypt(string pass)
    {
        string encryptedPass="";
        for (int i=0; i<pass.Length; i++)
        {
            char curChar = pass[i];
            if (IsDigit(curChar))
                encryptedPass += GetPrevDigit(curChar);
            else if (IsLetter(curChar))
                encryptedPass += GetNextLetter(curChar);
        }
        return encryptedPass;
    }
    // הפעולה מקבלת סיסמה מוצפנת ומחזירה סיסמה מופענחת
    public static string decrypt(string pass)
    {
        string decryptedPass="";
        for (int i=0; i<pass.Length; i++)
        {
            char curChar = pass[i];
            if (IsDigit(curChar))
                decryptedPass += GetNextDigit(curChar);
            else if (IsLetter(curChar))
                decryptedPass += GetPrevLetter(curChar);
        }
        return decryptedPass;
    }
    // הפעולה מקבלת סיסמה ומחזירה "אמת" אם הסיסמה חוקית ו"שקר" אחרת
    public static bool IsLegal(string pass)
    {
        bool hasLetter=false, hasDigit=false;
        int len = pass.Length;
        if (len<6 || len>8)
            return false;
        for (int i=0; i<pass.Length; i++)
        {
            char curChar = pass[i];
            if (IsDigit(curChar))
                hasDigit = true;
            else if (IsLetter(curChar))
                hasLetter = true;
            else
                return false;
        }
        return (hasLetter && hasDigit);
    }
    // פעולות עזר פרטיות

```

```

private static bool IsDigit(char ch)
{
    return (ch>='0' && ch<='9');
}
private static char GetPrevDigit(char ch)
{
    if (ch=='0')
        return '9';
    return (char)(ch-1);
}
private static char GetNextDigit(char ch)
{
    if (ch=='9')
        return '0';
    return (char)(ch+1);
}
private static bool IsLetter(char ch)
{
    return ( (ch>='a' && ch<='z') || (ch>='A' && ch<='Z') );
}
private static char GetNextLetter(char ch)
{
    if (ch=='z')
        return 'a';
    if (ch=='Z')
        return 'A';
    return (char)(ch+1);
}
private static char GetPrevLetter(char ch)
{
    if (ch=='a')
        return 'z';
    if (ch=='A')
        return 'Z';
    return (char)(ch-1);
}
}

```

## שאלה 2

לפניכם מחלקה המגדירה ארנק. בארנק יש מטבעות של 10 ש"ח, 5 ש"ח, ו-1 ש"ח. כדי שהארנק לא יהיה כבד, נגרום למספר המטבעות בו להיות מינימלי. למשל, אם בארנק יש 21 שקלים, הארנק יכיל שתי מטבעות של עשרה שקלים, אפס מטבעות של 5 שקלים ומטבע אחד של שקל אחד.

המחלקה מכילה את הפעולות הבאות:

◆ **הפעולה הבונה** – פעולה המאתחלת את הארנק להיות ריק.

◆ **GetAmount** – פעולה המחזירה מספר שלם המייצג את סכום הכסף שבארנק בשקלים.

◆ **AddMoney** – פעולה המקבלת סכום כסף שיש להוסיף לארנק ומעדכנת את מספר המטבעות בו מכל סוג.

◆ **Pay** – פעולה המקבלת סכום כסף לתשלום. הפעולה מפחיתה מהארנק את הסכום המבוקש ומשאירה את הארנק במספר מינימלי של מטבעות כפי שתואר לעיל. הפעולה מחזירה ערך בוליאני המציין אם היה די כסף בארנק.

השלימו את תכונות ופעולות המחלקה שלהלן:

```
/* מחלקת ארנק */
public class Wallet
{
    private int _____;
    private int _____;
    private int _____;
    public Wallet()
    {
        _____ = _____;
        _____ = _____;
        _____ = _____;
    }
    // הפעולה מחזירה את סכום הכסף שבארנק בשקלים
    public int GetAmount()
    {
        return _____;
    }
    // הפעולה מקבלת סכום כסף שיש להוסיף לארנק
    public void AddMoney(int givenSum)
    {
        int sum = GetAmount() + givenSum;
        _____ = _____ + sum / 10;
        sum = sum % 10;
        _____ = _____ + _____;
        sum = _____;
        _____ = _____ + _____;
    }
    // הפעולה מקבלת סכום כסף לתשלום,
    // הפעולה מחזירה ערך בוליאני המציין אם היה די כסף בארנק
    public bool Pay(int sumToPay)
    {
        if (_____)
            return false;
    }
}
```

```

        AddMoney(_____);
        return _____;
    }
} // class Wallet

```

## תשובה 2

```

/* מחלקת ארנק */
public class Wallet
{
    private int coins1;
    private int coins5;
    private int coins10;
    public Wallet()
    {
        coins1 = 0;
        coins5 = 0;
        coins10 = 0;
    }
    // הפעולה מחזירה את סכום הכסף שבארנק בשקלים
    public int GetAmount()
    {
        return coins10*10 + coins5*5 + coins1;
    }
    // הפעולה מקבלת סכום כסף שיש להוסיף לארנק
    public void AddMoney(int givenSum)
    {
        int sum = GetAmount() + givenSum;
        coins10 = sum / 10;
        sum = sum % 10;
        coins5 = sum / 5;
        sum = sum % 5;
        coins1 = sum;
    }
    // הפעולה מקבלת סכום כסף לתשלום,
    // הפעולה מחזירה ערך בוליאני המציין אם היה די כסף בארנק
    public bool Pay(int sumToPay)
    {
        if (GetAmount() < sumToPay)
            return false;
        AddMoney(sumToPay * -1);
        return true;
    }
} // class Wallet

```

### שאלה 3

מפעל הפיס מתכנן הגרלה חדשה. בהגרלה זו יונפקו מספר קבוע של כרטיסים, וסכום הפרסים הכולל יהיה 1,000,000 ₪. לכל כרטיס יהיה מספר מזל בן 8 ספרות (ייתכנו מספר כרטיסים שלהם אותם מספרי מזל). כרטיס זוכה יהיה כרטיס אשר סכום ארבע הספרות הראשונות של מספר המזל בו יהיה שווה לסכום ארבע הספרות האחרונות. בסיום הנפקת הכרטיסים ייקבע גודל הזכייה לכרטיס יחיד, כיוון שרק אז ידעו את מספר הכרטיסים הזוכים. אם כלל לא הונפקו כרטיסים זוכים, סכום הזכייה לכרטיס יהיה 0. לפניכם מחלקה המגדירה את מכוונת הנפקת הכרטיסים. השלימו את פעולות המחלקה שלפניכם:

```
using System;
/* מחלקה להנפקת כרטיסי הגרלה */
public class TicketFactory
{
    // הגדרת התכונות
    private const int TOTAL_PRIZE = _____;
    private int ticketCounter;
    private int winTicketCounter;
    // פעולה בונה
    public TicketFactory(int totalNumOfTickets)
    {
        ticketCounter = totalNumOfTickets;
        winTicketCounter = 0; // מספר הכרטיסים הזוכים
    }
    // פעולה המגרילה מספר מזל חדש
    public int GetNextLuckyNum()
    {
        if (ticketCounter <= 0)
            return 0;
        ticketCounter--;
        Random rnd = new Random();
        int num = _____; // הגרלת מספר מזל בן 8 ספרות
        if (_____)
            winTicketCounter++;
        return num;
    }
    // פעולה המחזירה את סכום הזכייה של כרטיס זוכה
    public double GetWinningPrize()
    {
        if (winTicketCounter == 0)
            return _____;
        else
            return _____;
    }
    // פעולה פרטית הבודקת אם מספר מזל הוא זוכה
    private bool IsWinnerNum(int num)
    {
        return (GetDigitsSum(_____) ==
                GetDigitsSum(_____));
    }
}
```



```

פעולה פרטית המחזירה את סכום ספרות המספר
private int GetDigitsSum(int num)
{
    int sum = _____;
    while (_____)
    {
        _____;
        _____;
    }
    return _____;
}
}

```

### תשובה 3

```

using System;
/* מחלקה להנפקת כרטיסי הגרלה */
public class TicketFactory
{
    // הגדרת התכונות
    private const int TOTAL_PRIZE = 1000000;
    private int ticketCounter;
    private int winTicketCounter;
    // פעולה בונה
    public TicketFactory(int totalNumOfTickets)
    {
        ticketCounter = totalNumOfTickets;
        winTicketCounter = 0; // מספר הכרטיסים הזוכים
    }
    // פעולה המגרילה מספר מזל חדש
    public int GetNextLuckyNum()
    {
        if (ticketCounter <= 0)
            return 0;
        ticketCounter--;
        Random rnd = new Random();
        // הגרלת מספר מזל בן 8 ספרות
        int num = rnd.Next(10000000,100000000);
        if (IsWinnerNum(num))
            winTicketCounter++;
        return num;
    }
    // פעולה המחזירה את סכום הזכייה של כרטיס זוכה
    public double GetWinningPrize()
    {
        if (winTicketCounter == 0)
            return 0;
        else
            return (double)TOTAL_PRIZE/winTicketCounter;
    }
    // פעולה פרטית הבודקת אם מספר מזל הוא זוכה

```

```

private bool IsWinnerNum(int num)
{
    return (GetDigitsSum(num/10000) ==
           GetDigitsSum(num%10000));
}
// פעולה פרטית המחזירה את סכום ספרות המספר
private int GetDigitsSum(int num)
{
    int sum = 0;
    while (num>0)
    {
        sum += num%10;
        num /= 10;
    }
    return sum;
}
}

```

## שאלה 7

פתחו אלגוריתם המקבל כקלט מספר שלם חיובי  $N$ , המבטא את מספר הבנים ואת מספר הבנות בכיתה ( $N$  בנים ו- $N$  בנות); ואחריו רשימה של  $2N$  מספרים.  $N$  המספרים הראשונים מייצגים עבור כל בן באיזו בת הוא בוחר, ו- $N$  המספרים הבאים מייצגים עבור כל בת באיזו בן היא בוחרת. האלגוריתם נותן כפלט את מספר הזוגות התואמים, כלומר מספר הזוגות של בן ובת שבחרו זה בזה.

לדוגמה, הקלט הבא: 3 3 1 1 2 2 1 3 מציין כי בכיתה יש 3 בנים ו-3 בנות (המספר הראשון בקלט הוא 3). שלושת המספרים הבאים מפרטים את בחירת הבנים. בן מספר 1 בחר בבת מספר 3. בן מספר 2 בחר בבת מספר 1 וגם בן מספר 3 בחר בבת מספר 1. שלושת המספרים האחרונים מפרטים את בחירת הבנות: בת מספר 1 בחרה בבן מספר 2. בת מספר 2 בחרה בבן מספר 2 ובת מספר 3 בחרה בבן מספר 1.

אם כך, יש שני זוגות אשר בחרו זה בזה: בן מספר 1 בחר בבת מספר 3, וגם היא בחרה בו, וכך גם בן מספר 2 ובת מספר 1 בחרו זה בזה. ממשו את האלגוריתם בשפת Java.

**הדרכה:** כתבו מחלקת "צמדים" אשר מגדירה שני מערכים: אחד עבור בחירת הבנים והאחר עבור בחירת הבנות. הגדירו במחלקה פעולה הסורקת את שני המערכים ומחזירה את מספר הזוגות התואמים.

## תשובה 7

רעיון הפתרון:

נסרוק את מערך הבנים, ועבור כל בן נבדוק האם הבת שאותה בחר, בחרה אותו.

יישום האלגוריתם:

```

/* מחלקת העדפות בנים בנות */
public class Preference
{
    // תכונות
    private int[] boys;

```

```

private int[] girls;

public Preference(int[] boysArr, int[] girlsArr)
{
    boys = boysArr;
    girls = girlsArr;
}

public int GetMatchCount()
{
    int count = 0;
    for (int boy = 0; boy < boys.Length; boy++)
    {
        if (boy == girls[boys[boy]])
            count++;
    }
    return count;
}

}

/* המחלקה הראשית */
using System;
public class UsePreference
{
    public static void Main()
    {
        Console.WriteLine("Enter number of boys and girls");
        int num = int.Parse(Console.ReadLine());
        int[] boys = new int[num];
        int[] girls = new int[num];

        Console.WriteLine("Enter the boys data:");
        for (int i = 0; i < num; i++)
        {
            Console.WriteLine("Enter preference of num {0}", (i+1));
            boys[i] = int.Parse(Console.ReadLine())-1;
        }

        Console.WriteLine("Enter the girls data:");
        for (int i = 0; i < num; i++)
        {
            Console.WriteLine("Enter preference of num {0}", (i+1));
            girls[i] = int.Parse(Console.ReadLine())-1;
        }

        Preference p = new Preference(boys, girls);
        Console.WriteLine("number of match preferences: {0}",
            p.GetMatchCount());
    }
}

```

## שאלה 8

בהשראת בגרות 2003 תשס"ג, שאלה 10  
לקראת תחרות ארצית במדעי המחשב, נערכה בחינת מיון ל-1750 תלמידים. לתחרות הארצית  
יתקבלו תלמידים שציונם בבחינת המיון גבוה מהציון הממוצע של כל הנבחנים בבחינה זו.  
א. כתבו מחלקת "תלמיד" המכילה את התכונות הבאות: שם, כתובת, מספר תעודת זהות, שפת  
התכנות המועדפת על התלמיד (C# או Java) וציון התלמיד בבחינת המיון.  
ב. כתבו פעולה ראשית הקולטת את נתוני כל המועמדים, ומציגה כפלט שתי רשימות המכילות  
את שמות התלמידים שיתקבלו לתחרות הארצית, את כתובותיהם ואת מספרי תעודת הזהות  
שלם. הרשימה הראשונה תכלול את פרטי התלמידים ששפת התכנות המועדפת עליהם היא  
C#, והרשימה השנייה תכלול את אלה שמעדיפים את Java.  
השתמשו במחלקה שכתבתם בסעיף א.

**הדרכה:** הפעולה הראשית צריכה לשמור עצמים מסוג תלמיד לצורך הצגת הפלט המבוקש.  
חשבו כיצד כדאי לשמור את הנתונים. להלן כמה אפשרויות:  
(1) לשמור את התלמידים במערך משותף לפי סדר קליטתם.  
(2) לשמור את התלמידים בשני מערכים לפי שפת התכנות המועדפת עליהם. האם ניתן לדעת  
מראש כמה מקום להקצות לכל מערך?  
(3) לשמור את התלמידים במערך משותף בהפרדה בין שתי הקבוצות (אלה המעדיפים את C#  
ואלה המעדיפים את Java) קבוצה אחת תמלא את המערך מההתחלה לכיוון הסוף והקבוצה  
השנייה תמלא את המערך מהסוף להתחלה.

## תשובה 8

```
/* מחלקת תלמיד מועמד */
class Student
{
    // תכונות
    private string name;
    private string address;
    private string id;
    private string favoriteLanguage;
    private double grade;

    public Student(string name, string id, string address,
                   string language, double grade)
    {
        this.name = name;
        this.id = id;
        this.address = address;
        this.favoriteLanguage = language;
        this.grade = grade;
    }

    public string GetFavoriteLanguage()
    {
        return this.favoriteLanguage;
    }

    public double GetGrade()
    {
        return this.grade;
    }
}
```

```

    }
    public string GetDetails()
    {
        return "name: "+name+" address: "+address+" id: "+id;
    }
} // class Student
/* מחלקת תחרות */
using System;
public class Competition
{
    public static void Main()
    {
        const int STUDENTS_NUM = 5;
        Student[] students = new Student[STUDENTS_NUM];
        int csi = 0;
        int javai = students.Length-1;

        for(int i=0; i<STUDENTS_NUM; i++)
        {
            Console.WriteLine("Student number {0}", (i+1));
            Console.Write("Enter Name: ");
            string name = Console.ReadLine();
            Console.Write("Insert ID: ");
            string id = Console.ReadLine();
            Console.Write("Enter Address: ");
            string address = Console.ReadLine();
            Console.Write("Enter Your Favorite Language: ");
            string language = Console.ReadLine();
            Console.Write("Enter Your Grade: ");
            double grade = double.Parse(Console.ReadLine());
            if (language == "C#")
            {
                students[csi] = new Student(name, id,
                    address, language, grade);
                csi++;
            }
            else
            {
                students[javai] = new Student(name, id,
                    address, language, grade);
                javai--;
            }
            Console.WriteLine();
        }

        double sum = 0;
        for(int i = 0; i < students.Length; i++)
            sum += students[i].GetGrade();
        double gradeAverage = sum/students.Length;

        Console.WriteLine("The list of the C# students:");
        for(int i = 0; i < csi; i++)
            if (students[i].GetGrade() > gradeAverage)

```

```
        Console.WriteLine(students[i].GetDetails());

    Console.WriteLine("The list of the Java students: ");
    for(int i = csi; i < students.Length; i++)
        if (students[i].GetGrade() > gradeAverage)
            Console.WriteLine(students[i].GetDetails());
    } // Main
} // class Competition
```