

## פרק 13 – פתרון בעיות

בפרק 11 התעמקנו בנושא מחלקות ועצמים ובפרק 12 התמקדנו במבנה נתונים מסוג מערך דו-ממדי ובפיתוח אלגוריתמים בסיסיים לחיפוש, למיון ולמיזוג. במהלך פתרון הבעיות התנסונו הן בהגדרת מחלקות, תכונות ופעולות והן בפיתוח ויישום של אלגוריתמים. בפרק זה נעסוק בבעיות המשלבות פיתוח אלגוריתמי יחד עם הגדרת מחלקות מתאימות לפתרון הבעיה. במהלך פתרון הבעיות נתמקד בתכונות ובפעולות הנדרשות ונתאים לכל בעיה את האלגוריתמים המתאימים לפתרונה.

בפרק זה תמצאו בעיות הלקוחות מפרק ג של בחינת הבגרות, ובעיות חדשות ברוח הפרק. בפרק זה בבגרות על הנבחן לנתח בעיה אלגוריתמית ולממש אותה באמצעות מחלקה מתאימה.

### הצ'יה 1 אפ'רות 2004 תש"ד, אלה 10

מטרת הבעיה ופתרונה: הצגת חיפוש במערך דו-ממדי.

נתון מערך דו-ממדי בגודל  $N \times N$  המכיל מספרים שלמים מ-1 עד  $MAX\_NUM$  (כולל). נגדיר כי במערך קיימת "רביעייה  $k$ ", אם המספר  $k$  מופיע ב-4 תאים של תת-מערך בגודל  $2 \times 2$ . לדוגמה, לפניכם מערך דו-ממדי בגודל  $5 \times 5$ . במערך יש "רביעייה 9":

1	3	2	2	8
2	9	9	6	1
12	9	9	1	4
17	6	8	5	2
1	3	5	7	1

הגדירו מחלקה המכילה מערך דו-ממדי כתכונה, וכללו בה את הפעולות הבאות:

- פעולה למציאת "רביעייה  $k$ ". הפעולה תקבל את המספר  $k$  ותחזיר `true` אם קיימת "רביעייה  $k$ " במערך ו-`false` אחרת.
- פעולה להחזרת המספר  $k$  הגדול ביותר שעבורו קיימת "רביעייה  $k$ " במערך. אם לא נמצאה רביעייה  $k$  במערך, תחזיר הפעולה -1. למימוש פעולה זו יש להשתמש בפעולה שכתבתם בסעיף הקודם.

בעיה זו נדרשו להגדיר מחלקה המכילה מערך דו-ממדי כתכונה ושתי פעולות הפועלות על המערך. בשלב ראשון, נגדיר בצורה מדויקת כיצד ניצג את התכונות וכיצד נממש כל פעולה. לאחר מכן, נממש את המחלקה בשפת `C#`.

### הגדרת המחלקה רביעייה $K$

#### הגדרת התכונות

- ♦ `matrix` – מערך דו-ממדי של מספרים מטיפוס שלם.
- ♦ `MAX_NUM` – שלם קבוע, המייצג את הערך המקסימלי שיכול להופיע במערך.

## הגדרת הפעולות

נפרט את כל פעולות המחלקה ונסביר את הפעולות הדורשות התייחסות מיוחדת:

♦ **הפעולה הבונה** – פעולה זו תקבל מערך דו-ממדי המשמש לאתחול matrix.

♦ **מצא רביעייה k** – פעולה זו תקבל מספר k ותחזיר true אם קיימת עבורו רביעייה במערך ו-false אחרת. כדי לממש פעולה זו עלינו לבדוק עבור כל תא במערך, אם הוא ושלושת שכניו (השכן מימינו, השכן מתחתיו וכן השכן האלכסוני בכיוון למטה וימינה) זהים למספר k. לשם כך, נשתמש בלולאה מקוננת, ועבור כל תא במערך שערך שווה ל-k נבדוק אם שכניו גם שווים ל-k (פרט לתאים בשורה האחרונה ובעמודה האחרונה):

```
if(matrix[i,j] == k)
{
    if ((matrix[i+1,j] == k) && (matrix[i,j+1] == k) &&
        (matrix[i+1,j+1] == k))
    {
        // מצאנו רביעייה k
    }
}
```

♥ **שימו**: אנו לא בודקים את התאים בשורה האחרונה ובעמודה האחרונה, כדי להימנע מגלישה מגבולות המערך.

♦ **מצא k מקסימלי** – פעולה זו תחזיר את המספר המקסימלי שקיימת עבורו "רביעייה k". יישום פעולה זו יתבצע באופן הבא: כיוון שאנו יודעים כי איברי המערך הינם מספרים שלמים בין 1 ל-MAX\_NUM. אנו יכולים לבדוק את כל המספרים באמצעות שימוש בפעולה מצא רביעייה k, החל מ-MAX\_NUM בסדר יורד עד 1. ברגע שנמצא מספר שקיימת עבורו "רביעייה k" נוכל לסיים את החיפוש כיוון שבידינו המספר הגדול ביותר העונה על הדרישה.

♥ **שימו**: האלגוריתם שהוצג לעיל מתאים למקרים ש-MAX\_NUM הוא מספר קטן יחסית, מדוע? חשבו על אלגוריתם חלופי למקרה ש-MAX\_NUM הוא מספר גדול.

## מימוש המחלקה

```
/*
    המחלקה רביעייה k
*/
public class QuadrupletK
{
    // הגדרת התכונות
    private int[,] matrix; // המערך
    private const int MAX_NUM = 30;
    // פעולה בונה
    public QuadrupletK(int[,] mat)
    {
        matrix = mat;
    } // QuadrupletK
    // בדיקה אם קיימת במערך רביעייה k
    public bool Find4K(int k)
    {
        int i,j;
        for (i = 0; i < matrix.GetLength(0) - 1; i++)
            for (j = 0; j < matrix.GetLength(1) - 1; j++)
```

```

        if (matrix[i,j] == k)
            if ( (matrix[i+1,j] == k) &&
                (matrix[i,j+1] == k) &&
                (matrix[i+1,j+1] == k) )
                return true;
            return false;
    } // Find4K
    // מציאת הרביעייה הגדולה ביותר במטריצה
    public int FindBiggestK()
    {
        int num = MAX_NUM;
        while (num > 0)
        {
            if (Find4K(num))
                return num;
            else
                num--;
        } // while
        return -1;
    } // FindBiggestK
} // class QuadrupletK

```

### סוף פתרון בעיה 1

**שימו ♥ :** בפעולה הבונה של המחלקה QuadrupletK אתחלנו את התכונה matrix באמצעות ההשמה matrix = mat; ולא באמצעות יצירת מערך חדש ובהעתקת האיברים אחד אחד. כתוצאה מהשמה זו התכונה matrix והמשתנה שהועבר כפרמטר מפנים לאותו מערך דו-ממדי. כעת, כל שינוי שיתבצע במערך דרך המשתנה שנמצא מחוץ לפעולה ישתקף גם במערך matrix. הדבר אינו רצוי במקרים שחשוב להגן על נתוני התכונה (על המערך matrix) מפני שינוי לא מבוקר מבחוץ. במקרה זה העדפנו לוותר על ההגנה ולבצע השמה פשוטה החוסכת הן מקום בזיכרון והן זמן בפעולת ההעתקה.

### שאלה 13.1 בהשראת בגרות 2003 תשס"ג, שאלה 9

בהינתן מערך דו-ממדי המכיל מספרים שלמים שונים זה מזה ובהינתן מספר b, נגדיר "תת-מערך b" כתת-המערך המתקבל מהתחום שמימין ומתחת למספר b במערך כולל המספר b. לדוגמה: עבור המערך הנתון ועבור המספר b=4:

2	7	12	3	17
27	22	4	0	1
9	-2	8	13	-9
-1	5	-20	20	10

ה"תת-מערך 4" הוא:

4	0	1
8	13	-9
-20	20	10

- א. הגדירו מחלקה בשם "תת-מערך מספרי" המכילה מערך דו-ממדי כתכונה (ובו מספרים שונים זה מזה), ופעולה בונה המקבלת מערך דו-ממדי המשמש לאתחול התכונה.
- ב. בהינתן מערך דו-ממדי כפי שתואר לעיל, פתחו אלגוריתם אשר ימצא את ה"תת-מערך k" עבור k נתון. אם האלגוריתם מצא את ה"תת-מערך k" המבוקש הוא יציג את איבריו, אחרת תוצג הודעה ש"תת-מערך k" לא נמצא.
- ג. כתבו פעולה במחלקה "תת-מערך מספרי" המממשת את האלגוריתם מהסעיף הקודם. הפעולה מקבלת כפרמטר מספר כלשהו k ומציגה את ה"תת-מערך k" שנמצא או הודעה ש"תת-מערך k" לא נמצא.
- הדרכה:** לצורך מימוש הפעולה, היעזרו בפעולת עזר פרטית המקבלת מציינים של איבר במערך הדו-ממדי ומציגה את איברי התת-מערך שנמצאים מימין ומתחת לאיבר המצוין, כולל האיבר עצמו.
- ד. הגדירו במחלקה פעולה נוספת המציגה את ה"תת-מערך k" עבור k שהוא המספר המינימלי במערך הדו-ממדי. הפעולה לא מקבלת פרמטרים, היא מוצאת את האיבר המינימלי במערך הדו-ממדי ולאחר מכן נעזרת בפעולה הפרטית מהסעיף הקודם לצורך הצגת איברי התת-מערך.

### שאלה 13.2 בהשראת בגרות 2005 תשס"ה, שאלה 9

נגדיר "פרח" במערך באופן הבא: הפרח מורכב מ-5 איברים בתת-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-ממדי המייצג שדה. הפעולה הבונה מקבלת מערך דו-ממדי, ומאתחלת בו את השדה.
- ב. כתבו פעולה המקבלת שני מספרים שלמים המציינים מיקום של איבר במערך (אינדקסים), המספר הראשון מצוין שורה, והמספר השני מצוין עמודה. הפעולה תחזיר true אם איבר זה הוא "לב הפרח" של "פרח" במערך; אחרת – הפעולה תחזיר false.
- ג. מערך הוא "פרחוני" אם יש בו לפחות 5 "פרחים".  
 כתבו פעולה שתבדוק אם המערך הוא "פרחוני", ותחזיר true או false 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.

## ח'יה 2 ההשראת הטרות 2004 תס"ד, אלה 9

מטרת הבעיה ופתרונה: הצגת בעיה מורכבת שלפתרונה יש להשתמש במערך של עצמים, במערך כתכונה של עצם ובתבנית מיון.

בסוכנות הנסיעות "שלום שלום" מארגנים טיולים. הסוכנות מציעה 100 טיולים, ולכל אחד מהם מספר בין 1 ל-100. לכל טיול יכולים להירשם עד 50 נוסעים. נוסע יכול להירשם לטיול רק אם יש מקום בטיול.

פתחו וממשו אלגוריתם לניהול סוכנות הנסיעות. האלגוריתם ירשום אנשים על פי שמם לטיולים שאליהם הם מבקשים להירשם, עד אשר יתקבל השם End לסיום הקלט. האלגוריתם יציג את רשימת הטיולים ממוינת לפי מספר האנשים בכל טיול, ויוסיף בצד כל טיול את רשימת שמות האנשים המשתתפים בו.

בעיה זו אנו נדרשים לרשום אנשים לטיול על בסיס מקום פנוי, ולהציג את כל הטיולים ממוינים לפי מספר הנרשמים אליהם. אם כך, בעיה זו עוסקת באוסף טיולים ומתאים לייצג אותנו כמערך. מידע על טיול כולל נתונים שונים (כגון מספר המשתתפים ורשימת המשתתפים). על כל טיול ניתן לבצע מספר פעולות שונות (כגון רישום משתתף). לכן מתאים להגדיר מחלקה בשם טיול.

### הגדרת המחלקה טיול

#### הגדרת התכונות

נבחן מהן הדרישות עבור עצם מהמחלקה טיול: יש לשמור את מספרו הסידורי של הטיול, את מספר המשתתפים בו ואת רשימת שמות האנשים הרשומים אליו. אם כך, תכונות של עצם מסוג "טיול" הן:

- ♦ `tripNum` – מספר שלם המייצג את מספרו הסידורי של הטיול.
- ♦ `numOfPassengers` – מספר שלם המייצג את מספר הנרשמים לטיול.
- ♦ `passengers` – מערך של מחרוזות המייצג את רשימת שמות הנרשמים לטיול.
- ♦ `MAX_PASS_PER_TRIP` – קבוע שלם שערכו 50 והוא מייצג את מספר הנוסעים המקסימלי בטיול.

#### הגדרת הפעולות

קעת נבחן מהן הפעולות שניתן לבצע על טיול: הוספת משתתף (אם יש מקום פנוי), והצגת רשימת שמות המשתתפים. אם כך פעולות של עצם מסוג "טיול" הן:

- ♦ **הפעולה הבונה** – פעולה זו תקבל כפרמטר את מספרו הסידורי של הטיול (ערך שלם), ותאתחל את התכונה `tripNum`. בנוסף, הפעולה הבונה תקצה את המערך של שמות המשתתפים ותאפס את התכונה של מספר המשתתפים.
- ♦ **הוספת מטייל** – פעולה זו תקבל את שם המטייל כפרמטר. אם יש מקום פנוי בטיול, הוא יצורף לרשימת המשתתפים, מספר המשתתפים יקודם ב-1 ויוחזר הערך `true`. אם אין מקום פנוי יוחזר הערך `false`.
- ♦ **החזרת רשימת המטיילים** – פעולה זו תחזיר את רשימת המטיילים לצורך תצוגה.

♦ החזרת מספרו הסידורי של הטיול – פעולה זו תחזיר את מספרו הסידורי של הטיול לצורך תצוגה.

♦ החזרת מספר המשתתפים בטיול – פעולה זו תחזיר את מספר המשתתפים בטיול לצורך מיון הטיולים על פי מספר המשתתפים.

## מימוש המחלקה

```
/*
    מחלקת טיול
*/
public class Trip
{
    // הגדרת התכונות
    private const int MAX_PASS_PER_TRIP = 50; // קבוע: מספר נוסעים
    private int tripNum; // מספר טיול
    private int numOfPassengers; // מספר נרשמים
    private string[] passengers; // שמות הנרשמים לטיול
    // פעולה בונה
    public Trip(int tripNum)
    {
        passengers = new string[MAX_PASS_PER_TRIP];
        this.tripNum = tripNum;
        numOfPassengers = 0;
    }
    // פעולות גישה
    public int GetTripNum()
    {
        return tripNum;
    }
    public int GetNumOfPassengers()
    {
        return numOfPassengers;
    }
    // הפעולה מחזירה את שמות המשתתפים בטיול
    public string[] GetPassengers()
    {
        string[] pass = new string[numOfPassengers];
        for(int i = 0; i < numOfPassengers; i++)
            pass[i] = passengers[i];
        return pass;
    }
    // הפעולה מוסיפה נוסע לטיול
    // ומחזירה ערך אמת אם יש מקום בטיול ושקר אחרת
    public bool AddPassenger(string passenger)
    {
        if (numOfPassengers < MAX_PASS_PER_TRIP)
        {
            passengers[numOfPassengers] = passenger;
            numOfPassengers++;
            return true;
        } // if
        return false;
    }
}
```

```

    } // AddPassenger
} // class Trip

```

**שימו** ♥ הפעולה GetPassengers יוצרת מערך חדש בשם pass וגודלו כמספר המשתתפים בטיול. הפעולה מעתיקה את רשימת המשתתפים בטיול לתוך המערך ומחזירה את המערך החדש. כלומר הפעולה מחזירה **עותק** של המערך ולא את המערך עצמו. חשוב, מה היה יכול לקרות אילו החזרנו פשוט את המערך passangers באופן הבא:

```

return passangers;

```

## הגדרת הפעולה הראשית

### פירוק לתת-משימות

משימות הפעולה הראשית הן:

4. יצירת מערך של 100 טיולים ואתחול הטיולים במערך.
5. קליטת שמות אנשים, ועבור כל אחד קליטת מספר הטיול שהוא מעוניין להירשם אליו ורישום לטיול זה.
6. מיון הטיולים לפי מספר המשתתפים. המיון שנשתמש בו הוא "מיון בועות".
7. הצגת רשימת הטיולים הממוינת.

### בחירת משתנים

בפעולה הראשית יוגדר מערך של 100 טיולים, כלומר של עצמים מהמחלקה טיול.

### מימוש הפעולה הראשית

```

/*
    המחלקה הראשית
*/
using System;
public class TravelAgency
{
    public static void Main()
    {
        // הגדרת והקצאת משתנים
        const int TRIPS_NUM = 100; // קבוע: מספר טיולים
        Trip[] trips = new Trip[TRIPS_NUM]; // הקצאת מערך הטיולים
        string passengerName; // שם נוסע
        int tripNum; // מספר טיול מבוקש
        // הקצאת הטיולים
        for (int i = 0; i < TRIPS_NUM; i++)
            trips[i] = new Trip(i + 1);
        Console.WriteLine("Enter passenger name, type 'End' to " +
            "finish: ");
        passengerName = Console.ReadLine();
        while (passengerName!="End")
        {
            Console.WriteLine("Enter the trip number: ");
            tripNum = int.Parse(Console.ReadLine());
            if (trips[tripNum-1].AddPassenger(passengerName))
                Console.WriteLine("You were added " +
                    "successfully");
            else

```

```

        Console.WriteLine("This trip is full");
        Console.Write("Enter passenger name, type 'End' " +
            "to finish: ");
        passengerName = Console.ReadLine();
    } // while
    // חיון הטיולים לפי כמות הנרשמים
    Trip temp;
    for (int i = 1; i < TRIPS_NUM; i++)
    {
        for (int j = 0; j < TRIPS_NUM - i; j++)
        {
            if (trips[j].GetNumOfPassengers() >
                trips[j+1].GetNumOfPassengers())
            {
                // החלפה
                temp = trips[j];
                trips[j] = trips[j+1];
                trips[j+1] = temp;
            } // if
        } // for j
    } // for i
    string[] passengers;
    // הצגת הטיולים והמשתתפים בהם, לפי מספר המשתתפים
    for (int i = 0; i < TRIPS_NUM ; i++)
    {
        Console.WriteLine("Trip {0} has {1} passengers",
            trips[i].GetTripNum(),
            trips[i].GetNumOfPassengers());
        passengers = trips[i].GetPassengers();
        for (int j = 0; j < passengers.Length; j++)
        {
            Console.WriteLine(passengers[j]);
        }
    } // for
} // main
} // class TravelAgency

```

## סוף פתרון מציה 2

### שאלה 13.3

באולימפיאדת הארנבים מתקיימת תחרות בשלוש קטגוריות שונות: ריצה לגזר, ריצה לכרוב וריצה לברוקולי. באולימפיאדה מחולקות שתי מדליות זהב. את הראשונה מקבל הארנב שממוצע התוצאות שלו בשלוש התחרויות הוא הטוב ביותר (הנמוך ביותר). את השנייה מקבל הארנב שזמן הריצה שלו הוא הנמוך ביותר מבין כל זמני הריצה של כל הארנבים בכל הקטגוריות כולן. פתחו וממשו אלגוריתם העוזר למארגני התחרות לבחור את המנצחים. האלגוריתם יקלוט תחילה את מספר הארנבים המתחרים, לאחר מכן יקלוט רשימה של שמות הארנבים ואת זמני הריצה לכל ארנב בכל קטגוריה. פלט האלגוריתם יהיה שמות שני הזוכים במדליות הזהב.

**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" הן: שם הארנב ומערך המכיל את שלוש 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 ותשמור אותם במערך ארנבים. היא תחפש את הארנב שיש לו התוצאה הממוצעת הנמוכה ביותר ואת הארנב שיש לו התוצאה הנמוכה ביותר, ותציג את שמם.

### בצ'יה 3

מטרת הבעיה ופתרונה: הצגת שילוב תבניות

נצחיה המורה להיסטוריה מעוניינת לדעת כמה תלמידים קיבלו את הציון הגבוה ביותר (שאינו בהכרח 100) במבחן השכבתי. צחי הכין עבור נצחיה את קטע התוכנית הבא:

```
int maxCount=0;
int max=0;
int numOfStudents;
int grade;
Console.Write("Enter number of students: ");
numOfStudents = int.Parse(Console.ReadLine());
for (int i = 0; i < numOfStudents; i++)
{
    grade = int.Parse(Console.ReadLine());
    if (grade > max)
        max = grade;
    if (grade == max)
        maxCount++;
}
Console.WriteLine("{0} students received the maximum grade",
                    maxCount);
```

א. האם קטע התוכנית משיג את מטרתו?

ב. אם קטע התוכנית שגוי הציעו תיקון כך שהוא ישיג את מטרתו עבור כל קלט אפשרי.

### בדיקת קטע התוכנית

בעיה זו אנו נדרשים לנתח קטע תוכנית ולבחון אם הוא משיג את מטרתו. לשם כך, עלינו לבחון דוגמאות קלט שונות ולבדוק אם פלט התוכנית תקין. לשם בדיקת התוכנית ניתן להסתפק בדוגמאות שבהן 10 תלמידים בלבד.

#### שאלה 13.4

הריצו את התוכנית ובדקו עבור דוגמאות הקלט הבאות אם הפלט תקין (משמאל לימין המספר הראשון בקלט מייצג את מספר התלמידים):

א. 10 90 55 84 90 74 89 80 84 90 56

ב. 10 84 71 84 90 53 84 76 90 85 67

בבדיקה זו עבור דוגמת קלט א, נקבל פלט תקין המודיע כי שלושה תלמידים קיבלו את הציון המקסימלי. לעומת זאת, עבור דוגמת קלט ב נקבל פלט שגוי המודיע כי ארבעה תלמידים קיבלו את הציון המקסימלי ולא שניים כפי שניתן לראות מהקלט.

### מציאת השגיאה ותיקונה

כעת כדי לתקן את השגיאה עלינו להבין מדוע שגיאה זו התרחשה. כפי שניתן לראות בדוגמת הקלט הראשונה, הציון המקסימלי מתקבל ראשון. המשתנה max שתפקידו לשמור את הציון

הגבוה ביותר מקבל את הערך 90. המשתנה `maxCount` שתפקידו למנות כמה פעמים מתקבל הציון הגבוה ביותר מקבל את הערך 1, כיוון שעד כה הציון 90 התקבל פעם אחת. מרגע זה ואילך, כל הציונים שמתקבלים, קטנים או שווים ל-`max`. במקרה שציון קטן מ-`max`, לא מתבצע דבר. במקרה שציון שווה ל-`max`, המשתנה `maxCount` מתקדם ב-1. כך התוכנית ממשיכה עד לסיום הקטע בהצלחה.

לעומת זאת, בדוגמת הקלט השנייה הציון המקסימלי לא מתקבל הראשון. תחילה מתקבל הציון 84 אשר נמנה פעמיים כציון הגבוה ביותר, ורק לאחר מכן מתקבל הציון 90. ברגע שמתקבל הציון 90 המשתנה `max` מתעדכן ל-90 (הגבוה מ-84), והמשתנה `maxCount` מתקדם כי התקבל ציון מקסימלי אחד נוסף. כאן נמצא שורש הבעיה, אף על פי שהמשתנה `max` עודכן למקסימום החדש, המשתנה `maxCount` לא אופס, וכך נמשיך למנות באופן שגוי בכל פעם החל מהערך שנמצא ב-`maxCount`.

המסקנה המתבקשת היא, שעלינו לאפס את המונה `maxCount` ברגע שמעדכנים את המשתנה `max`. נעשה זאת ב-`if` הראשון, שבודקים בו אם התקבל ציון גבוה יותר מהציון שנשמר עד כה. אם אכן התקבל ציון גבוה יותר, יש לאפס את המונה שסופר כמה פעמים התקבל ציון זה. אם כך, קטע התוכנית ייראה כך לאחר התיקון:

```
int maxCount=0;
int max=0;
int numOfStudents;
int grade;
Console.WriteLine("Enter number of students: ");
numOfStudents = int.Parse(Console.ReadLine());
for (int i = 0; i < numOfStudents; i++)
{
    grade = int.Parse(Console.ReadLine());
    if (grade > max)
    {
        max = grade;
        maxCount = 0; // איפוס המונה maxCount
    }
    if (grade == max)
        maxCount++;
}
Console.WriteLine("{0} students received the maximum grade",
maxCount);
```

### סוף פתרון בעיה 3

בעיה 3 נעזרנו בשתי תבניות מוכרות: תבנית מקסימום ותבנית מנייה. התבניות שזורות זו בזו כך שברגע שמשתנה המקסימום מוחלף, יש לאתחל מחדש את המונה. לפעמים משולבות יחדיו תבניות שונות בקטע קוד אחד. במקרים כאלה יש להקפיד על שילוב נכון של חלקי התבניות.

**שימו** ♥: כאשר משלבים כמה תבניות בפתרון בעיה, עלולים להשמיט חלק מההוראות הנדרשות או לכתוב אותן שלא במקומן. יש להקפיד ולבדוק היטב שהרכבת התבניות נעשתה בהצלחה, ולא נגרמו שגיאות לקוד עקב הרכבה לא מוצלחת.

### שאלה 13.5

נצחיה, המורה להיסטוריה, מעוניינת לדעת את ממוצע כל אחת מ-6 הכיתות שלה במבחן המשווה, וכן את ממוצע 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.WriteLine("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?

### שאלה 13.6

במפעל מועסקים 100 עובדים. בעל המפעל מעוניין לדעת נתונים על המשכורות שעליו לשלם בסופו של חודש. כל עובד נמצא במפעל 25 ימים בחודש ועובד בכל יום מספר מסוים של שעות. התעריף לעובדים אינו קבוע אלא אישי ומשתנה מעובד לעובד.

פתחו אלגוריתם הקולט עבור כל אחד מ-100 העובדים את שמו, את התעריף שלו לשעה, וכמה שעות עבד בכל אחד מ-25 ימי העבודה באותו חודש. האלגוריתם יציג כפלט את סכום המשכורות שעל בעל המפעל לשלם לעובדיו. ממשו את האלגוריתם **בשתי הדרכים** הבאות:

א. ממשו את האלגוריתם כולו בפעולה הראשית, כך שלולאה חיצונית תקלוט נתונים לכל אחד מ-100 העובדים. בכל סיבוב של הלולאה ייקלט שם עובד ותעריף לשעה ולאחר מכן בלולאה פנימית, ייקלטו השעות שהעובד עבד בכל אחד מ-25 ימי העבודה בחודש. עליכם לצבור עבור כל עובד את מספר השעות שעבד, ולהכפילן בתעריף שלו. בנוסף עליכם לצבור את המשכורות של כל העובדים.

**שימו ♥:** זכרו לאפס את צובר השעות של העובד במקום המתאים.

ב. הגדירו מחלקת עובד. תכונות עובד כוללות: שם, תעריף לשעה וסכום השעות שהעובד עבד במשך החודש. למחלקה זו הגדירו פעולה בונה המקבלת את שם העובד ואת התעריף לשעה. בנוסף כתבו פעולה המקבלת את השעות שהעובד עבד באחד מהימים ומעדכנת את סכום השעות החודשי של העובד, ופעולה המחזירה את המשכורת החודשית שיש לשלם לעובד. בפעולה הראשית, ממשו לולאה אשר תייצר עצם מסוג עובד לכל אחד מ-100 העובדים,

ובלולאה פנימית תקלוט ותעדכן את מספר השעות שהעובד עבד בכל אחד מ-25 ימי העבודה. סכמו את המשכורות של כל 100 העובדים.

**שימו** ♥: היכן איפסתם את צובר השעות? האם בגרסה השנייה של פתרון הבעיה קיים החשש כי נשכח לאפס את הצובר?

## קצ'ה 4

**מטרת הבעיה ופתרונה:** הצגת מקסימום כתכונה, הצגת מונה כתכונה והצגת מערך מונים כתכונה.

טורניר השש-בש מתחיל ועדיין לא נמצאו הקוביות! כתבו מחלקה המגדירה זוג קוביות שש-בש תקניות. בנוסף לערכי הקוביות, עצם מסוג זוג קוביות יכול מידע סטטיסטי על זוג הקוביות. כדי לדמות את הטלת הקוביות ואת החזרת ערכי הקוביות, ממשו במחלקה את הפעולות:

```
public void ThrowDice ()
public int GetDie1 ()
public int GetDie2 ()
```

בסיום המשחק, נוכל לקבל מעצם מסוג "זוג קוביות" את המידע הבא:

א. מה סכום ההטלה הגבוה ביותר שהתקבל בזוג הקוביות במשחק?

ב. כמה פעמים הוגרל דאבל בזוג הקוביות?

ג. מהי תוצאת הדאבל שהוגרלה הכי הרבה פעמים? אם הוגרלה יותר מאחת כזו תוחזר התוצאה שערכה גדול יותר.

## הגדרת המחלקה זוג קוביות

בעיה זו אנו נדרשים לדמות הטלת שתי קוביות ובמקביל לאסוף מידע סטטיסטי על ההטלות. לשם כך נגדיר את המחלקה "זוג קוביות". עצם מסוג זוג קוביות יאפשר לנו להטיל את הקוביות ולשמור מידע סטטיסטי הקשור להטלתן.

## הגדרת התכונות

נבחן מהם הנתונים שעלינו לשמור כדי שנוכל לענות על כל אחת מהשאלות המבוקשות:

א. מה סכום ההטלה הגבוה ביותר שהתקבל בזוג הקוביות במשחק?

כדי לענות על שאלה זו נשתמש בתבנית **מקסימום** לחישוב הערך הגבוה ביותר שהתקבל עד רגע ההטלה. נגדיר משתנה מקסימום כתכונה של העצם, ולאחר כל הטלה הוא יעודכן לפי הצורך.

ב. כמה פעמים הוגרל דאבל בזוג הקוביות?

כדי לענות על שאלה זו נשתמש בתבנית **מנייה**. משתנה המנייה יישמר כתכונה, ולאחר כל הטלה יעודכן משתנה זה לפי הצורך.

ג. מהי תוצאת הדאבל שהוגרלה הכי הרבה פעמים?

לשאלה זו שש תוצאות אפשריות שונות. (דאבל-1 הוגרל הכי הרבה פעמים, דאבל-2 הוגרל הכי הרבה פעמים, ..., דאבל-6 הוגרל הכי הרבה פעמים). כדי לענות על שאלה זו עלינו למנות עבור כל אחת מ-6 התוצאות האפשריות כמה פעמים היא התקבלה, ולבדוק איזו תוצאה התקבלה הכי הרבה פעמים. לצורך כך נשתמש בתבנית **מערך-מונים**. מערך המונים יישמר כתכונה, ולאחר כל הטלה הוא יעודכן לפי הצורך.

אם כך, התכונות של עצם מסוג "זוג קוביות" הן :

- ◆ **die1** – מספר שלם המייצג את תוצאת ההטלה האחרונה של הקובייה הראשונה.
- ◆ **die2** – מספר שלם המייצג את תוצאת ההטלה האחרונה של הקובייה השנייה.
- ◆ **maxDiceSum** – מספר שלם המייצג את הסכום המקסימלי שהוגרל בהטלת הקוביות עד כה.
- ◆ **doubleCount** – מספר שלם המייצג את מספר הפעמים שהתקבל דאבל עד כה.
- ◆ **doubleCountArr** – מערך בגודל 6 מטיפוס שלם המייצג כמה פעמים הוגרל כל אחד מהדאבלים האפשריים.

### הגדרת הפעולות

כעת נבחן מהן הפעולות שניתן לבצע על עצם מסוג "זוג קוביות".

- ◆ **הפעולה הבונה** – הפעולה תאפס את התכונות, תקצה מקום למערך מוני הדאבל ותאפס אותו.
- ◆ **הטל קוביות** – הפעולה תגריל שני מספרים אקראיים בין 1 ל-6 ותציב אותם בתכונות המתאימות.
- ◆ **החזרת תוצאת ההטלה של הקובייה הראשונה** – הפעולה תחזיר את תוצאת ההטלה של הקובייה הראשונה.
- ◆ **החזרת תוצאת ההטלה של הקובייה השנייה** – הפעולה תחזיר את תוצאת ההטלה של הקובייה השנייה.
- ◆ **החזרת סכום ההטלה הגבוה ביותר** – הפעולה תחזיר את הסכום המקסימלי שהתקבל בהטלת הקוביות.
- ◆ **החזרת מספר דאבלים** – הפעולה תחזיר את מספר הפעמים שהתקבל דאבל.
- ◆ **החזרת מספר דאבלים שהתקבל הכי הרבה פעמים** – הפעולה תסרוק את מערך המונים ותחזיר את המספר שהוגרל הכי הרבה פעמים כדאבל. אם הוגרלו כמה תוצאות כאלה, תוחזר הגדולה מביניהן. אם לא התקבלו דאבלים כלל יוחזר הערך 0.

### מימוש המחלקה

```
/* מחלקת זוג קוביות */
using System;
public class Dice
{
    // הגדרת התכונות
    private int die1; // תוצאת ההטלה של הקובייה הראשונה
    private int die2; // תוצאת ההטלה של הקובייה השנייה
    private int maxDiceSum;
    private int doubleCount;
    private int[] doubleCountArr; // מערך המונים עבור הדאבלים
    // פעולה בונה
    public Dice()
    {
        die1 = 0;
        die2 = 0;
        maxDiceSum = 0;
        doubleCount = 0;
    }
}
```

```

        doubleCountArr = new int[6];
        for (int i = 0; i < 6; i++)
            doubleCountArr[i] = 0;
    }
    //פעולות גישה
    public int GetDie1()
    {
        return die1;
    }
    public int GetDie2()
    {
        return die2;
    }
    // הפעולה מטילה את שתי הקוביות ומעדכנת את התכונות
    public void ThrowDice()
    {
        Random rnd = new Random();
        die1 = rnd.Next(1,7);
        die2 = rnd.Next(1,7);
        // עדכון הסכום המקסימלי שהתקבל מהטלת הקוביות
        if (die1 + die2 > maxDiceSum)
            maxDiceSum = die1 + die2;
        // עדכון כמות הפעמים שהתקבל דאבל, ומערך המונים
        if (die1 == die2)
        {
            doubleCount++;
            doubleCountArr[die1-1]++;
        }
    }
    // החזרת סכום הטלת זוג הקוביות הגבוה ביותר שהיה במשחק
    public int GetMaxDiceSum()
    {
        return maxDiceSum;
    }
    // החזרת מספר הפעמים שהתוצאה היתה דאבל
    public int GetDoubleCount()
    {
        return doubleCount;
    }
    // הפעולה מחזירה את תוצאת הדאבל שהתקבלה הכי הרבה פעמים
    public int GetMaxTimesDouble()
    {
        int maxIndex = 0;
        if (doubleCount == 0) // לא היה אף דאבל
            return 0;
        for (int i = 1; i < doubleCountArr.Length; i++)
            if (doubleCountArr[i] > doubleCountArr[maxIndex])
                maxIndex = i;
        return maxIndex + 1;
    }
}
} // class Dice

```

**סוף פתרון בעיה 4**

### שאלה 13.7

משרד החינוך החליט לבדוק את הרגלי הקריאה של הנוער. הוחלט לבצע סקר, כך שכל משתתף בו ירשום את שמו, ויקליד 1 עבור כל ספר שקרא מרשימת 50 הספרים שהוצגו לפניו או 0 אם לא קרא את הספר. כתבו תוכנית שתקלוט את תשובותיהם של משתתפי הסקר. הקליטה תסתיים עם קבלת "@@@" (כשם הקורא). על התוכנית להציג את התשובות לשאלות:

- מספר המשתתפים שלא קראו אף אחד מהספרים שברשימה.
- מספר המשתתפים שקראו יותר ממחצית הספרים שברשימה.
- עבור כל אדם שקרא את כל הספרים יש להדפיס את שמו בצירוף הודעה "נוער למופת".
- מספר הספר הנפוץ ביותר (מספר הספר שקראו מספר הקוראים הגדול ביותר).

**הדרכה:** כתבו מחלקה המגדירה "סקר". הגדירו לעצם מסוג סקר את התכונות הבאות המייצגות את תוצאות הסקר: מספר המשתתפים שלא קראו אף ספר מהרשימה, מספר המשתתפים שקראו יותר ממחצית הספרים שברשימה ומערך מונים המונה עבור כל ספר את מספר המשתתפים שקראו אותו.

הגדירו במחלקה "סקר" את הפעולות הבאות:

פעולה בוליאנית המקבלת מערך בגודל 50 המייצג תשובות של קורא יחיד לסקר. הפעולה מעדכנת את התכונות המייצגות את תוצאות הסקר ומחזירה true אם התלמיד קרא את כל הספרים אחרת יחזר false.

הוסיפו פעולות המחזירות את מספר המשתתפים שלא קראו אף ספר מהרשימה, את מספר המשתתפים שקראו יותר ממחצית הספרים שברשימה ואת מספר הספר הנפוץ ביותר.

### שאלה 13.8

מנהלי קבוצת הכדורסל "ג'ירפות 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 1-1 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 C# 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.

### שאלה 13.9 בהשראת בגרות 2006 תשס"ו, שאלה 10

בית ספר מזמין מחנות ספרים ספרי קריאה עבור 620 תלמידיו.

הספרים בחנות מסומנים בקודים. קוד יכול להיות מספר בין 1 ל-315. אם יש בחנות כמה עותקים מאותו ספר, הם מסומנים באותו קוד. כל תלמיד מזמין לפחות ספר אחד, ומחיר כל ספר לתלמיד הוא 28 שקל. בית הספר גובה מהתלמידים את התשלום בעבור הספרים שהזמינו, ומעביר את התשלום הכולל לחנות הספרים.

- א. פתחו אלגוריתם שיקלוט את הזמנות התלמידים ויטפל בתשלומי התלמידים ובתשלום בית הספר, ממשו אותו בשפת C#. עליכם לפתח את האלגוריתם לפי השלבים שלפניכם:
- i. הגדירו את מחלקת "הזמנות ספרים". עצם מסוג זה יכול את סך כל ההזמנות שביצעו התלמידים עבור כל ספר וספר.
  - ii. הגדירו במחלקת "הזמנות ספרים" פעולה "הזמנה מתלמיד". פעולה זו תקבל מערך המכיל את קודי הספרים שהזמין תלמיד ותעדכן את ההזמנות. הפעולה תחזיר את הסכום שעל התלמיד לשלם עבור הזמנה זו.
  - iii. הגדירו במחלקת "הזמנות ספרים" פעולה "סכום הזמנות". פעולה זו תכין מחרוזת המפרטת עבור כל קוד של ספר את מספר העותקים שהוזמנו ממנו בסך הכול. כמו כן, יחושב ויפורט התשלום הכולל שעל בית הספר להעביר לחנות בעבור כל הספרים שהוזמנו.
- ב. כתבו בשפת C# את הפעולה הראשית אשר תנהל את ההזמנות. בפעולה הראשית ייקלטו מספר הספרים שכל תלמיד מזמין; ואחריו ייקלטו הקודים של הספרים שהתלמיד מזמין. בסיום הזמנה של תלמיד יוצג כפלט הסכום שעליו לשלם בעבור הספרים שהזמין. בסיום קליטת הנתונים של כל התלמידים יוצג סיכום ההזמנות.
- הערה:** אין צורך לבדוק את תקינות הקלט.

## קצ'ה 5

**מטרת הבעיה ופתרונה:** יחסים בין מערכים.

נתונה סדרה של מספרים, עליכם לפתח אלגוריתם המפצל את הסדרה לשתי סדרות, האחת מכילה את המספרים הזוגיים והשנייה את האי-זוגיים. הגדירו מחלקה בשם EvenOdd, במחלקה יהיו שלושה מערכים: המערך השלם המכיל את כל המספרים ושני מערכים נוספים: אחד עבור המספרים הזוגיים והאחר עבור המספרים האי-זוגיים. בנוסף תהיה פעולה המפצלת את המערך השלם לשני מערכים – אחד של מספרים זוגיים והאחר של מספרים אי-זוגיים.

### הגדרת המחלקה EvenOdd

בבעיה זו אנו נדרשים לפתח אלגוריתם המפצל מערך אחד לשני מערכים שונים.

#### הגדרת התכונות

התכונות של עצם מסוג EvenOdd כוללות שלושה מערכים: המערך השלם, מערך הזוגיים ומערך האי-זוגיים.

#### הגדרת הפעולות

פעולות של עצם מסוג EvenOdd הן: הפעולה הבונה המקבלת את המערך השלם, פעולה המפצלת את המערך השלם לשני מערכים ושתי פעולות גישה – אחת מחזירה את מערך הזוגיים והאחרת מחזירה את מערך האי-זוגיים.

נבחן את האלגוריתם הדרוש לביצוע פעולת הפיצול:

כדי לדעת את הגודל הדרוש עבור כל אחד מהמערכים החדשים, עלינו תחילה לסרוק את המערך השלם, למנות את מספר הזוגיים ואת מספר האי-זוגיים, ורק לאחר מכן נוכל להקצות את המערכים החדשים בהתאם לגודל הנדרש.

לשם הפיצול עלינו לסרוק את המערך השלם ובו-בזמן למלא את המערכים החדשים בהתאם.



**שימו** ♥: בתהליך הפיצול עלינו להשתמש ב-3 אינדקסים שונים. אחד עבור המערך השלם, והוא יתקדם לתא הבא בכל סבב של הלולאה. בנוסף, לכל אחד משני המערכים החדשים יהיה אינדקס אשר יקודם ב-1 רק במקרה הצורך.

### מימוש המחלקה

```
/*
 מחלקה לפיצול מערך שלם לזוגיים ולא-זוגיים
*/
public class EvenOdd
{
    // הגדרת התכונות
    private int[] arr; // המערך השלם
    private int[] oddArr; // מערך המספרים הזוגיים
    private int[] evenArr; // מערך המספרים האי-זוגיים
    // פעולה בונה
    public EvenOdd(int[] a)
    {
        arr = a;
    }
    // פיצול
    public void Split()
    {
        // סופרים כמה זוגיים יש במערך
        int evenCount = 0;
        for (int i = 0; i < arr.Length; i++)
            if (arr[i] % 2 == 0)
                evenCount++;
        // מקצים מקום עבור מערך הזוגיים
        evenArr = new int[evenCount];
        // כעת נוכל להסיק כמה אי זוגיים יש במערך
        oddArr = new int[arr.Length - evenCount];
        int iEven = 0, iOdd = 0;
        for (int i = 0; i < arr.Length; i++)
        {
            if (arr[i] % 2 == 0)
            {
                evenArr[iEven] = arr[i];
                iEven++;
            }
            else
            {
                oddArr[iOdd] = arr[i];
                iOdd++;
            }
        }
    } // Split
    public int[] GetOddArr()
    {
        return oddArr;
    }
    public int[] GetEvenArr()
    {
```

```

        return evenArr;
    }
} // EvenOdd

```

### סוף פתרון בעיה 5

**שימו** ♥: בפעולה הבונה של מחלקה זו, כמו בפעולה הבונה של בעיה 1 בפרק זה, אתחלנו את התכונה `arr` באמצעות השמה פשוטה ולא באמצעות העתקה. גם כאן ויתרנו על ההגנה על המערך `arr` מפני שינויים לא מבוקרים מבחוץ, בתמורה לחיסכון במקום ובזמן.

### 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.

### בעיה 6

**מטרת הבעיה ופתרונה**: יחסים בין מערכים ושימוש בפעולת עזר פרטית.

פתחו אלגוריתם שיגדיר מערך דו-ממדי של מספרים שלמים בגודל  $N \times M$  (N שורות ו-M עמודות). האלגוריתם ייצור מערך חד-ממדי בן N איברים כך שבמיקום ה-k במערך החד-ממדי יהיה סכום איברי השורה ה-k במערך הדו-ממדי. למשל, אם המערך הדו-ממדי הוא:

1	3	2
3	1	5
5	6	2
3	7	2
7	6	3

אז המערך החד-ממדי יהיה:

6	9	13	12	16
---	---	----	----	----

במקום השלישי מוצב הערך 13, כי סכום האיברים בשורה השלישית במערך הדו-ממדי הוא 13. כתבו מחלקה ובה מוגדר מערך דו-ממדי כתכונה ומוגדרת פעולה המחזירה את מערך הסכומים.

### הגדרת המחלקה סכום-שורות

בעיה זו אנו נדרשים לפתח אלגוריתם הסוכם שורות של מערך דו-ממדי ומשים את התוצאות במערך חד-ממדי.

### הגדרת התכונות

התכונה של עצם מסוג סכום-שורות היא המערך הדו-ממדי.

## הגדרת הפעולות

הפעולות של עצם מסוג סכום-שורות הן: פעולה בונה המקבלת את המערך הדו-ממדי, ופעולה לסכימת השורות המחזירה את המערך החד-ממדי הנדרש.

נבחן את האלגוריתם שיסכום את השורות:

עלינו לעבור על המערך הדו-ממדי, שורה אחר שורה, לסכום את איברי השורה ולהציב את הסכום במיקום המתאים במערך החד-ממדי. לשם כך, נשתמש בפעולת עזר **פרטית** אשר תקבל מספר שורה ותחזיר את סכומה.

## מימוש המחלקה

```
/*
    מחלקה לסכימת שורות מטריצה
*/
public class LinesSum
{
    // הגדרת התכונות
    private int[,] mat;
    // פעולה בונה
    public LinesSum(int[,] a)
    {
        mat = a;
    }
    // פעולה פרטית לסכימת שורה במטריצה
    private int SumOneLine(int lineNum)
    {
        int sum = 0;
        for (int i = 0; i < mat.GetLength(1); i++)
            sum = sum + mat[lineNum,i];
        return sum;
    }
    // סכימת השורות והחזרת הסכומים במערך
    public int[] SumLines()
    {
        // מקצים מקום עבור מערך הסכומים
        int[] sumArr = new int[mat.GetLength(0)];
        for (int i = 0; i < mat.GetLength(0); i++)
            // קריאה לפעולה פרטית לקבלת הסכום
            sumArr[i] = SumOneLine(i);
        return sumArr;
    } // SumLines
} // LinesSum
```

**סוף פתרון קציה 6**

## 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
-----	----	-----	----	----

יחזור 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 יחזור המערך:

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

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

### שאלה 13.12

אופיר קיבלה מערכת שעות לשנת הלימודים החדשה, והחליטה לכתוב את המחלקה "מערכת שעות" אשר תעזור לה בניהול סדר היום שלה. המחלקה מכילה מערכת שעות כתכונה (הניחו כי אופיר לומדת 5 ימים בשבוע, 8 שעות בכל יום). בנוסף לפעולה הבונה המקבלת את מערכת השעות ומאתחלת אותה, במחלקה יוגדרו שלוש הפעולות הבאות: הפעולה הראשונה מקבלת מספר יום (1 עבור יום ראשון, 2 עבור יום שני וכך הלאה) ומחזירה מערך של מחרוזות המייצג את מערכת השיעורים עבור יום זה. הפעולה השנייה מקבלת יום ושעה (לדוגמה 1 1 ייצג את השיעור הראשון ביום ראשון) ותחזיר מחרוזת המייצגת את השיעור המתקיים במועד המבוקש. פעולה שלישית תקבל שם מקצוע (כמחרוזת) ותחזיר כמה פעמים בשבוע אופיר לומדת מקצוע זה. ממשו מחלקה זו בשפת C#.

## ביצה 7 מבטאות 2007 תשס"ז, אלה 9

**מטרת הבעיה ופתרונה:** הצגת בעיה שלפתרונה יש להשתמש בצובר ובחיפוש ערך במערך.

בעל חניון למכוניות החליט למחשב את ניהול החניון. בחניון יש 318 מקומות חניה, הממוספרים מ-1 עד 318. החניון פתוח בכל יום מהשעה 6:00 עד השעה 23:00. בחניון משלמים 14 שקל על כל שעת חניה. כלי רכב יכולים להיכנס או לצאת מהחניון רק בשעות שלמות. אפשר להיכנס לחניון עד השעה 22:00 (כולל). בסוף כל יום לא נשארות מכוניות בחניון.

א. פתחו אלגוריתם שיטפל בניהול החניון ביום מסוים. עליך לפתח את האלגוריתם לפי השלבים שלפניך:

- i בחרו במשתנים עיקריים, הגדירו את טיפוסיהם ותארו את תפקידיהם.
- ii פרקו את הבעיה לתת-משימות. באלגוריתם יש לכלול את התת-משימות האלה:
- \* פתיחת החניון בתחילת היום – איפוס הקופה וסימון כל מקומות החניה כפנויים.
- \* כניסת רכב לחניון – קליטת שעת הכניסה של הרכב (מספר שלם בין 6 ל-22 (כולל)), מציאת מקום פנוי לרכב, הדפסת המספר של המקום הפנוי, סימון מקום החניה כתפוס. אם אין מקום פנוי, תודפס הודעה מתאימה.
- \* יציאת רכב מהחניון – קליטת המספר של מקום החניה של הרכב, קליטת שעת היציאה שלו (מספר שלם בין 7 ל-23 (כולל)), חישוב והדפסה של התשלום, עדכון הקופה, סימון מקום החניה כפנוי.
- \* סגירת החניון בסוף היום – הדפסת סך כל הכסף שנגבה במשך היום בעבור חניית מכוניות בחניון.

הגדירו לכל אחת מהתת-משימות את מטרתה (טענת כניסה וטענת יציאה), וישמו כל אחת מהתת-משימות באמצעות פעולה בשפת C#.

ב. כתבו בשפת C# תוכנית לניהול החניון, שתיישם את האלגוריתם שפיתחתם בסעיף א. לאחר פתיחת החניון התוכנית תקלוט בעבור כל רכב: את הקוד 1 אם הרכב נכנס לחניון, ואת הקוד 2 אם הרכב יוצא מהחניון, ותבצע את התת-משימות בהתאם. הקליטה תסתיים כאשר ייקלט הקוד 1-. לאחר סיום הקליטה התוכנית תסגור את החניון. עליכם להשתמש בפעולות שיישמתם בסעיף א.

**הניחו:** שעת היציאה היא תמיד מספר גדול יותר משעת הכניסה.

**הערה:** אין צורך לבדוק את תקינות הקלט.

## הגדרת המחלקה חניון

### הגדרת התכונות

התכונות של עצם מסוג חניון כוללות מערך בגודל 318, אשר ייצג את מקומות החניה בחניון. תא שערכו 0 יחשב כמקום חניה פנוי. בתא תפוס תישמר שעת כניסת הרכב לחניון. בנוסף נשמור במשתנה מטיפוס שלם את קופת החניון אשר תצבור את תשלומי הרכבים בעת יציאתם מן החניון. אם כך, תכונות של עצם מסוג "חניון" הן:

◆ **spaces** – מערך של שלמים המייצגים את המקומות בחניון. גודל המערך 318.

◆ **cash** – מספר שלם הצובר את סכום הכסף בקופה.

◆ **NUM\_OF\_SPACES** – קבוע שלם שערכו 318 – מייצג את מספר המקומות בחניון.

◆ **HOURLY\_RATE** – קבוע שלם שערכו 14 – מייצג את המחיר לשעת חניה.

### הגדרת הפעולות

להלן פעולות העצם כפי שהן מפורטות בהגדרת השאלה:

◆ **הפעולה הבונה** – הפעולה תקצה מקום למערך spaces ותאתחל את התאים ב-0. בנוסף הפעולה תאפס את הסכום המצטבר בקופה.

◆ **כניסת רכב לחניון** – הפעולה תקבל כפרמטר את שעת הכניסה לחניון – מספר שלם בין 6 ל-22. הפעולה תבדוק במערך spaces אם יש מקום פנוי בחניון. אם יש מקום, שעת הכניסה תישמר בתא המתאים במערך והפעולה תחזיר את מספרו של המקום הפנוי שנמצא, אחרת יוחזר 0.

◆ **יציאת רכב מהחניון** – הפעולה תקבל כפרמטר את מספר מקום החניה – מספר שלם בין 1 ל-318, ואת שעת היציאה – מספר שלם בין 7 ל-23. הפעולה תסמן את המקום כפנוי, תחשב את הסכום לתשלום, תעדכן את הסכום המצטבר בקופה ותחזיר את הסכום לתשלום.

◆ **סגירת החניון** – הפעולה תחזיר את הסכום שהצטבר בקופה.

**שימו:** ♥ בחרנו לא לקלוט מידע או להציג אותו בתוך הפעולות השונות. במקום זאת, הגדרנו לכל פעולה פרמטרים שבאמצעותם היא מקבלת את פרטי המידע הנדרש לביצועה. כמו כן, כל פעולה מחזירה ערך המייצג את תוצאת ביצועה. נקלוט את הנתונים ונציג את התוצאות בפעולה הראשית.

## מימוש המחלקה

```
/*
    מחלקת חניון
*/
public class ParkingLot
{
    private int[] spaces;
    private int cash;
    private const int HOURLY_RATE = 14;
    private const int NUM_OF_SPACES = 318;
    // פעולה בונה, מאפסת את הקופה ואת המערך
    public ParkingLot()
    {
        cash = 0;
        spaces = new int[NUM_OF_SPACES];
        for (int i = 0; i < spaces.Length ; i++)
            spaces[i] = 0;
    }
    // פעולה המקבלת שעת כניסה ומחזירה את מספרו של המקום הפנוי הראשון,
    // אם אין מקום פנוי יוחזר 0
    public int CarIn(int hour)
    {
        int spaceNumber = 0;
        int i = 0;
        while (i < spaces.Length && spaceNumber == 0)
        {
            if(spaces[i] == 0)
            {
                spaceNumber = i + 1;
                spaces[i] = hour;
            }
            i++;
        }
        return spaceNumber;
    }
    // פעולה המקבלת שעת כניסה ומקום בחניון, ומחזירה את הסכום שיש לשלם
    public int CarOut(int spaceNumber, int hour)
    {
        int sumToPay;
        sumToPay = (hour - spaces[spaceNumber-1]) * HOURLY_RATE;
        spaces[spaceNumber-1] = 0;
        cash = cash + sumToPay;
        return sumToPay;
    }
    // פעולה המחזירה את הפדיון היומי
    public int EndOfDay()
    {
        return cash;
    }
} // class ParkingLot
```

## הגדרת הפעולה הראשית

### פירוק לתת-משימות

משימות הפעולה הראשית הן:

1. פתיחת החניון.
2. קליטת קודים: 1 לכניסת רכב, 2 ליציאת רכב, -1 לסיים יום פעילות.
3. סגירת החניון

### בחירת משתנים

בפעולה הראשית יוגדר עצם מטיפוס חניון ומשתנים עבור הפעולה המבוקשת, שעת הכניסה או היציאה, מקום החניה והסכום לתשלום.

### מימוש הפעולה הראשית

```
/*
        המחלקה הראשית
*/
using System;
public class ParkingManager
{
    public static void Main()
    {
        // הגדרת משתנים והקצאתם
        ParkingLot parkingLot = new ParkingLot(); // עצם מסוג חניון
        int action; // הפעולה המבוקשת
        int hour; // שעת כניסה או יציאה
        int numOfSpace; // מספר מקום החניה
        int sumToPay; // סכום לתשלום
        // קליטת הפעולה המבוקשת
        Console.WriteLine("Enter 1 for entry, 2 for exit, -1 to end: ");
        action = int.Parse(Console.ReadLine());
        while (action != -1)
        {
            if (action == 1) // כניסה לחניון
            {
                Console.WriteLine("Enter the time - " +
                    "a number between 6-22: ");
                hour = int.Parse(Console.ReadLine());
                numOfSpace = parkingLot.CarIn(hour);
                if (numOfSpace == 0)
                    Console.WriteLine("No free spaces");
                else
                    Console.WriteLine("Space number {0}", numOfSpace);
            }
            if (action == 2) // יציאה מהחניון
            {
                Console.WriteLine("Enter the time - " +
                    "a number between 7-23: ");
                hour = int.Parse(Console.ReadLine());
                Console.WriteLine("Enter your space number: ");
                numOfSpace = int.Parse(Console.ReadLine());
            }
        }
    }
}
```

```

        sumToPay = parkingLot.CarOut(numOfSpace, hour);
        Console.WriteLine("You need to pay {0}", sumToPay);
    }
    Console.Write("Enter 1 for entry, 2 for exit, -1 to"
        + " end: ");

    action = int.Parse(Console.ReadLine());
} // while
// סגירת החניון
Console.WriteLine("Total sum at the end of the day: {0}",
    parkingLot.EndOfDay());

} // Main
} // class ParkingManager

```

## סוף פתרון תרגיל 7

### תרגיל 8 מבחן 2007 תשס"ז, שאלה 10

**מטרת התרגיל:** הצגת בעיה שלפתרונה יש להשתמש בשילוב של תבניות אלגוריתמיות: מנייה, האם ערך בסדרה מקיים תנאי והזזה של תת-סדרה שמאלה.

א. איברים ברצף במערך חד-ממדי הם איברים שהאינדקסים שלהם הם מספרים עוקבים. כתבו בשפת C# פעולה בשם Seven שתקבל מערך חד-ממדי בגודל 105 המכיל מספרים שלמים. הפעולה תבדוק אם יש במערך 7 (או יותר) איברים ברצף שהערך של כל אחד מהם הוא אפס. אם כן הפעולה תחזיר "אמת" אחרת תחזיר "שקר".

ב. כתבו בשפת C# פעולה בשם Shift שתקבל מערך חד-ממדי a בגודל 105 המכיל מספרים שלמים, ומספר שלם k בין 1 ל-4 (כולל). הפעולה תבצע הזזה של כל אחד מאיברי המערך k מקומות שמאלה מהמקום שבו הוא נמצא. לאחר ההזזה האיברים השמאליים לא יופיעו יותר במערך. ב-k המקומות הימניים במערך יוצב 0.

דוגמה עבור  $k=2$  ומערך a בגודל 5:

3	7	2	0	1
---	---	---	---	---

ג. לאחר הפעלת הפעולה Shift המערך שיתקבל יראה כך:

2	0	1	0	0
---	---	---	---	---

ד. נתון מערך חד-ממדי בגודל 105 המכיל מספרים שלמים. כתבו בשפת C# תוכנית שתשתמש בפעולה Seven ותבדוק אם יש במערך 7 (או יותר) איברים ברצף שהערך של כל אחד מהם הוא 0. אם אין – התוכנית תקלוט מספר שלם k בין 1 ל-4 (כולל) ותבצע את הפעולה Shift עם המספר k שנקלט. התוכנית תמשיך לקלוט מספרים ולבצע עם כל אחד מהם את הפעולה Shift, עד שיהיו במערך 7 (או יותר) איברים ברצף שכל אחד מהם הוא 0. בסיום הביצוע התוכנית תדפיס את המערך.

**הערה:** אין צורך לבדוק את תקינות הקלט.

בסעיפים א ו-ב נתבקשו לכתוב פעולות המקבלות מערך כפרמטר. הפעולה הראשונה בודקת קיום תנאי כלשהו במערך והפעולה השנייה מבצעת הזזה לאיברי המערך. בסעיף ג עלינו לכתוב תוכנית המשתמשת בפעולות אלה.

מכיוון שנתבקשו לכתוב פעולות ולא להגדיר עצם כלשהו נוכל לכתוב אותן כפעולות סטטיות במחלקת שירות (כפי שהוצג בבעיה 7 בפרק 11).



## הגדרת המחלקה שירותי-מערך

כזכור, מחלקת שירות לא מגדירה תכונות כי תכונות שייכות לעצם ואנו לא יוצרים שום עצם. הפעולות הסטטיות פועלות על הפרמטרים שהן מקבלות ולא על תכונות. הפעלת הפעולות הסטטיות מתבצעת בדרך כלל ישירות דרך שם המחלקה ולא באמצעות עצם של המחלקה. כמו כן, לא נזדקק לפעולה בונה שתפקידה ליצור עצם ולאתחל את תכונותיו.

### הגדרת הפעולות

להלן פעולות השירות כפי שהן מפורטות בהגדרת השאלה:

♦ **Seven** – פעולה סטטית המקבלת מערך של שלמים כפרמטר ובודקת אם קיים בו רצף של שבעה אפסים או יותר. אם רצף כזה קיים הפעולה תחזיר **true** אחרת יוחזר **false**.

♦ **Shift** – פעולה סטטית המקבלת מערך של שלמים ומספר שלם  $k$  כפרמטר. הפעולה מזיזה את איברי המערך  $k$  מקומות שמאלה ומציבה אפס ב- $k$  האיברים האחרונים של המערך. פעולה זו אפשר לבצע כך: נעבור על כל איבר במערך פרט ל- $k$  הראשונים ונעביר אותו  $k$  מקומות שמאלה. לאחר ההזזה נשתמש בלולאה העוברת על  $k$  האיברים האחרונים של המערך ומציבה בהם אפס.

♥ **שימו**: מכיוון שהגישה למערך ב- $C\#$  נעשית דרך הפניה (כתובת בזיכרון), כל שינוי שהפעולה תבצע במערך שהתקבל כפרמטר יהיה למעשה שינוי במערך המקורי שהועבר לפעולה.

### מימוש המחלקה

```
/*
מחלקת שירותי מערך
*/
public class ArrayServices
{
    // הפעולה מקבלת כפרמטר מערך חד-ממדי
    // הפעולה מחזירה "אמת" אם קיים רצף של שבעה אפסים או יותר,
    // אחרת יוחזר "שקר"
    public static bool Seven(int[] a)
    {
        int counter = 0;
        for(int i = 0; i < a.Length; i++)
        {
            if (a[i] == 0)
                counter++;
            else
            {
                if (counter >= 7)
                    return true;
                else
                    counter = 0;
            }
        }
        if (counter >= 7)
            return true;
        else
            return false;
    }
}
```

```

// הזזת כל האיברים החל מהאיבר ה-k, k מקומות שמאלה
public static void Shift(int[] a, int k)
{
    for (int i = k; i < a.Length; i++)
        a[i-k] = a[i];
    for (int i = a.Length - k; i < a.Length; i++)
        a[i] = 0;
}
} //ArrayServices

```

את פעולת ה-Shift אפשר לבצע בכמה דרכים. דרך נוספת היא לבצע k פעמים הזזה אחת שמאלה של כל איברי המערך:

```

public static void Shift(int[] a, int k)
{
    for (int i = 0; i < k; i++)
        for (int j = 0; j < a.Length - 1; j++)
            a[j] = a[j+1];
    for (int i = a.Length - k; i < a.Length; i++)
        a[i] = 0;
}

```

## מימוש הפעולה הראשית

הפעולה הראשית תקלוט מערך של 105 שלמים ותבדוק באמצעות הפעולה Seven אם יש במערך לפחות 7 אפסים ברצף. אם אין – התוכנית תקלוט מספר שלם k בין 1 ל-4 ותבצע את הפעולה Shift עם המספר k שנקלט. התוכנית תמשיך לקלוט מספרים ותבצע עם כל אחד מהם את הפעולה Shift, עד שהפעולה Seven תחזיר true. בסיום הביצוע התוכנית תדפיס את המערך.

עד כה יצרנו מחלקה נפרדת עם הפעולה הראשית Main, והפעולות הסטטיות הופעלו בציון שם המחלקה, למשל כך: `ArrayServices.Shift(a, 3)`. קיימת גם אפשרות נוספת והיא לשים את הפעולה הראשית Main בתוך מחלקת השירות, יחד עם הפעולות Seven ו-Shift. במקרה זה נוכל להפעיל את הפעולות הסטטיות בצורה ישירה ללא קידומת שם המחלקה. התבוננו בפתרון הבא:

```

/*
    מחלקת שירותי מערך והפעולה הראשית
*/
using System;
public class ArrayServices
{
    public static bool Seven(int[] a)
    {
        ... // גוף הפעולה מופיע כאן
    }
    public static void Shift(int[] a, int k)
    {
        ... // גוף הפעולה מופיע כאן
    }
    // שימו לב לכך שהפעולה הראשית מזמנת את הפעולות הסטטיות
    // בצורה ישירה מבלי לציין את שם המחלקה כקידומת
    public static void Main()
    {
        int k;

```

```

int[] a = new int[105];
// קלט למערך
Console.WriteLine("Enter 105 array values: ");
for (int i = 0; i < a.Length; i++)
    a[i] = int.Parse(Console.ReadLine());
while (Seven(a) == false)
{
    Console.Write("Enter a number between 1-4: ");
    k = int.Parse(Console.ReadLine());
    Shift(a, k);
}
// הדפסת המערך
Console.WriteLine("The array values are: ");
for (int i = 0; i < a.Length; i++)
    Console.Write("{0} ", a[i]);
} //Main
} //ArrayServices

```

**סוף פתרון בציה 8**

## שאלות נוספות

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 האות a והספרה 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 תיחשב 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 אם הסיסמה חוקית
public static bool IsLegal(string pass)
{
    ...
}
}

```

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

```

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 מכונת הנפקת הכרטיסים. השלימו את פעולות המחלקה שלפניכם:

```

/* 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 _____;
    }
}

```

4. בחנות התקליטים הוחלט לבדוק איזה סוג מוזיקה הוא הנמכר ביותר. כל אריזות התקליטים בחנות מקודדות בקוד בן 2 ספרות. הספרה השמאלית מייצגת את המחלקה והספרה הימנית היא מספר התקליטים באריזה. המחלקות הקיימות בחנות:

1 – קלאסי	3 – ישראלי	5 – היפ הופ
2 – רוק	4 – ג'אז	6 – רגאי

לדוגמה, אריזה בקוד 62, מכילה תקליט כפול ממחלקת רגאי. פתחו וממשו אלגוריתם הקולט את קודי אריזות התקליטים שנמכרו במשך היום, ומציג כפלט באיזו מחלקה נמכר מספר התקליטים הגדול ביותר. הקלט מסתיים בקוד 0. שימו לב, תקליט כפול נחשב כשני תקליטים, תקליט משולש נחשב כשלושה וכן הלאה. הדרכה: כתבו מחלקה המגדירה חנות תקליטים. הגדירו תכונות לייצוג המידע הנדרש ופעולה המעדכנת את הנתונים לאחר כל מכירה.

5. בהשראת בגרות 2006 תשס"ו, שאלה 9

בהינתן מערך דו-ממדי שאיבריו הם המספרים 0 ו-1. נגדיר "שרשרת" במערך כרצף של איברים בשורה מסוימת או רצף של איברים בעמודה מסוימת המכילים את המספר 1. אורך "שרשרת" הוא מספר האיברים ב"שרשרת".

אם בשורה כלשהי או בעמודה כלשהי אין "שרשרת", אורך ה"שרשרת" בה יהיה 0. נתון כי בכל שורה או עמודה יכולה להיות לכל היותר "שרשרת" אחת.

איבר במערך ייקרא "מוקף", אם הוא מכיל את המספר 1 וגם אורך ה"שרשרת" בשורה שבה הוא נמצא שווה לאורך ה"שרשרת" בעמודה שבה הוא נמצא.

דוגמה: במערך בגודל 4x5 שלפניכם יש שני איברים "מוקפים".

0	0	0	1	0
0	1	1	1	0
0	0	0	1	0
0	0	0	0	1

"איבר מוקף" ←

← "איבר מוקף"

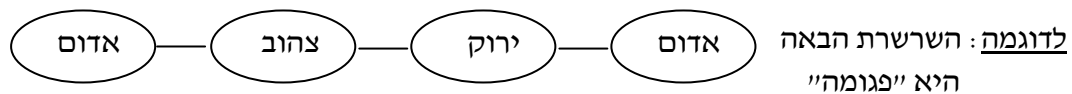
א. כתבו את המחלקה "שרשרת במערך" המכילה מערך דו-ממדי כתכונה, ובה פעולה בונה המקבלת מערך דו-ממדי ומאתחלת את התכונה. ניתן להניח שהמערך המתקבל כפרמטר תקין, כלומר מכיל רק את הערכים 0 ו-1, ושכל שורה או עמודה יש לכל היותר שרשרת אחת.

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

ג. כתבו פעולה שתמנה את מספר האיברים ה"מוקפים" שיש במערך, ותחזיר ערך בהתאם. השתמשו בפעולה שכתבתם בסעיף ב.

6. בהשראת בגרות 2005 תשס"ה, שאלה 10

במפעל לתכשיטים מרכיבים שרשרות מְחֻרוּזִים בשלושה צבעים: אדום, צהוב וירוק. בכל שרשרת יש לפחות חרוז אחד מכל צבע. שרשרת "אחידה" היא שרשרת שבה יש מספר שווה של חרוזים מכל אחד מהצבעים. שרשרת "פגומה" היא שרשרת שאינה "אחידה".



א. כתבו את המחלקה "מפעל תכשיטים". הגדירו פעולה המקבלת שרשרת ובודקת אם היא אחידה או פגומה. הגדירו פעולות המחזירות את המידע הבא: כמה שרשרות נוצרו באותו היום וכמה מתוכן היו פגומות.

ב. כתבו פעולה ראשית שקולטת את השרשרות שמיוצרות במפעל ביום מסוים. בעבור כל שרשרת יש לקלוט מחרוזת המורכבת מהאותיות B, G, R ו-Y המייצגות את הצבעים: אדום, ירוק, כחול וצהוב בהתאמה. הקלט עבור השרשרת שבדוגמה לעיל יהיה המחרוזת "RYGR". הקלט יסתיים במחרוזת ריקה "".

לאחר סיום הקלט יש להציג את מספר השרשרות שיוצרו במפעל באותו היום, ואת מספר השרשרות הפגומות. השתמשו במחלקה שכתבתם בסעיף א.

הערה: אין צורך בבדיקת תקינות הקלט.

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 : מצייין 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 C#.

**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.

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 שלהם. 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.

9. חנות הדיסקים "דיסקו" שכרה יועץ כדי לארגן מחדש את החנות. היועץ המליץ לארגן את קטלוג הדיסקים בחנות באופן הבא: היוצרים יהיו מסודרים בסדר אלפביתי ולכל יוצר תהיה רשימה ממוינת של שמות הדיסקים שלו. כתבו תוכנית הקולטת שם של יוצר ושם של דיסק ובודקת אם הוא נמצא בחנות.



**הדרכה :** כתבו את המחלקה "יוצר" המכילה שם של יוצר ומערך של שמות הדיסקים שלו. הפעולה הראשית תכיל מערך של יוצרים ותמייך אותם לפי שמותיהם. אין צורך לקלוט את שמות היוצרים ואת שמות הדיסקים שלהם. ניתן להניח שמערך היוצרים מאותחל בעצמים מסוג יוצר.

10. (הרחבת שאלה 13.6)

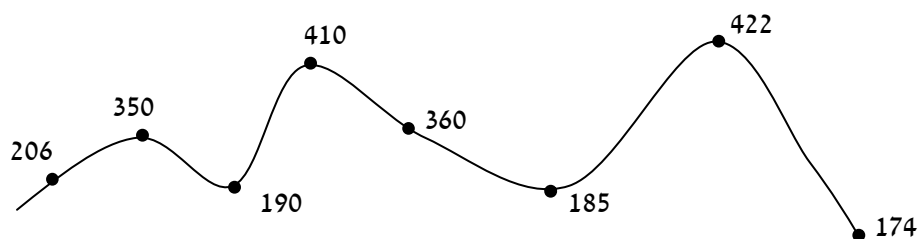
במפעל מועסקים 100 עובדים. בעל המפעל מעוניין לדעת נתונים על המשכורות שעליו לשלם בסופו של חודש מסוים. כל עובד נמצא במפעל 25 ימים בחודש ועובד מספר מסוים של שעות ביום. התעריף לעובדים אינו קבוע אלא אישי ומשתנה מעובד לעובד. פתחו וממשו אלגוריתם הקולט עבור כל אחד מ-100 העובדים את שמו, את התעריף שלו לשעה, וכמה שעות עבד בכל אחד מ-25 ימי העבודה באותו חודש. האלגוריתם יציג כפלט את הערכים הבאים :

- סכום המשכורות שעל בעל המפעל לשלם לעובדיו.
- המשכורת הגבוהה ביותר המשולמת באותו חודש ושמו של העובד שקיבל משכורת זו.
- מספר המשכורות מעל לממוצע.
- ממוצע שעות העבודה ליום לכל עובדי המפעל.
- שמות העובדים שעבדו פחות ממספר השעות הממוצע ביום במשך יותר מ-5 ימים במהלך החודש.

11. במגמת גיאוגרפיה נלמדו מספר הגדרות :

**פסגה** – נקודת גובה אשר מימינה ומשמאלה יש נקודות נמוכות יותר.  
**מדרון** – נקודת גובה אשר מצד אחד – נקודה גבוהה ממנה, ומצד שני – נמוכה ממנה.  
**מישור** – נקודת גובה אשר לפחות מצד אחד קיימת נקודה אשר שווה לה.  
**עמק** – נקודת גובה אשר מימינה ומשמאלה נקודות גבוהות יותר.  
 לאחר מכן יצאו לטיול לימודי שבו המדריך הכריז מידי פעם על גובה השטח. התלמידים נדרשו לרשום עבור כל נקודה (פרט לראשונה ולאחרונה) את הגדרתה הגיאוגרפית, ולספור את מספר הפסגות שעברו.

לדוגמה : עבור הנקודות בתרשים הבא המסודרות משמאל לימין יש לרשום : פסגה עמק פסגה מדרון עמק פסגה (לא נתייחס לנקודה הראשונה ולנקודה האחרונה) ונספרו בסך הכול 3 פסגות.



אורי החליט לכתוב תוכנית אשר מקבלת כקלט מספר המציין כמה נקודות גובה הוכרזו בטיול, ואחריו רשימה של נקודות הגובה. פלט התוכנית יהיה שעורי הבית שהוא נדרש להגיש.

**הדרכה:** הגדירו מחלקה בשם "מסלול טיול" המכילה את רשימת נקודות הגובה כתכונה. הגדירו פעולות המחזירות את המידע המבוקש.

## **סיכום**

בפרק זה הצגנו בעיות המשלבות פיתוח אלגוריתמי יחד עם הגדרת מחלקות מתאימות לפתרון הבעיות. במהלך פתרון בעיה זיהינו את התכונות הנדרשות לייצוג עצם מהמחלקה ואת הפעולות הנדרשות למימוש האלגוריתמים המתאימים לפתרונה. ראינו מקרים רבים שפעולה כלשהי נעזרה בפעולה נוספת (פרטית) לצורך מימוש האלגוריתם. בכל הבעיות והשאלות שהוצגו בפרק נעשה שימוש בתבניות ובאלגוריתמים שנלמדו בפרקים הקודמים. בחלק מהבעיות שילבנו יותר מתבנית אחת והדגשנו את המורכבות הנוספת הקיימת בשילוב זה.

בנוסף, הצגנו מגוון רחב של בעיות אשר הופיעו בפרק ג של בחינת הבגרות, משנת 2003 ועד שנת 2007, וכן בעיות נוספות ברוח פרק זה. לכל בעיה צירפנו הנחיות מתאימות לפתרונה, כדי להקנות לכם את הניסיון הנדרש בניתוח ובפתרון בעיות הכוללות פיתוח אלגוריתמי והגדרת מחלקות.

פרק זה מסכם את לימוד יסודות מדעי המחשב ומכין אתכם בהצלחה לקראת השלב הבא – עיצוב תוכנה.