

שיעור : 10/9

ראינו בשיעור הקודם שתוכנית מחשב בנויה מאוסף של הוראות , חד משמעיות , אשר באות לפי סדר מלמעלה למטה אחת אחרי השנייה . בשיעור שעבר בנינו משחק קטן בשפת סקראץ' .

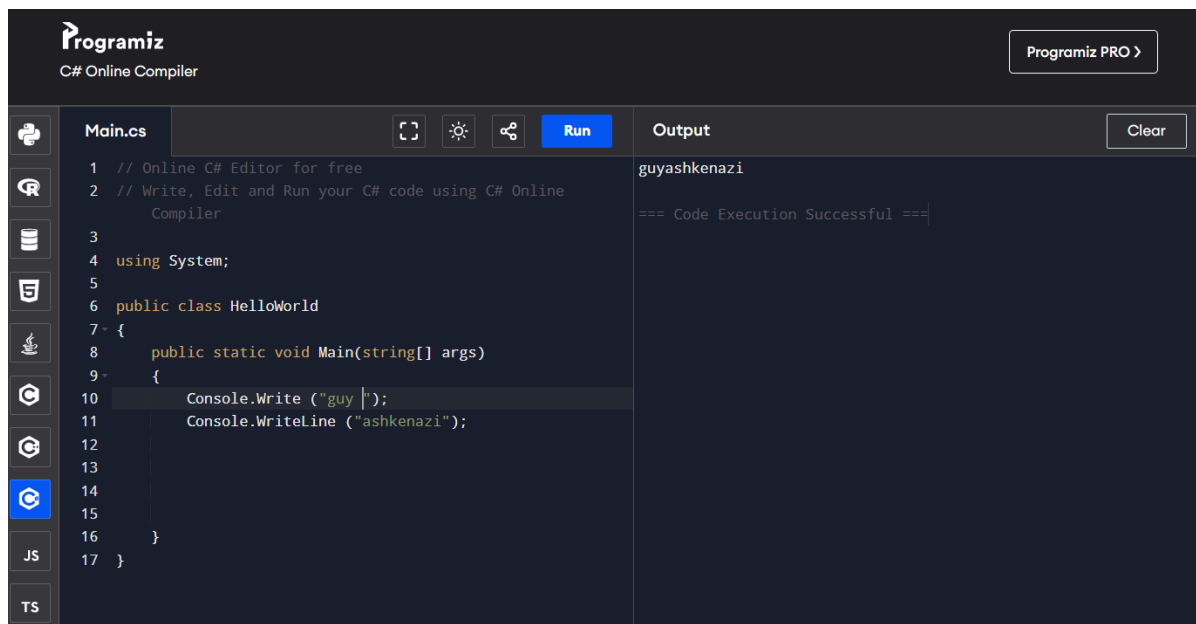
היום בשיעור התחלנו להכיר את מבנה שפת #C (סי שארפ) וראינו שמבנה התוכנית שלנו היא

```
using System;

public class HelloWorld
{
    public static void Main(string[] args)
    {
        Console.Write ("guy");
        Console.WriteLine ("ashkenazi");

    }
}
```

ראינו שכתובת התוכנית במחשב נעשת בעזרת קומפילר (מהדר) , אשר יודע לקבל את ההוראות שאנו כותבים למחשב ולהריץ/להפעיל אותם .
(על ידי לחיצה על מקש ה RUN) .



The screenshot shows the Programiz C# Online Compiler interface. The editor displays the following C# code:

```
1 // Online C# Editor for free
2 // Write, Edit and Run your C# code using C# Online
  Compiler
3
4 using System;
5
6 public class HelloWorld
7 {
8     public static void Main(string[] args)
9     {
10         Console.Write ("guy ");
11         Console.WriteLine ("ashkenazi");
12     }
13
14
15
16 }
17 }
```

The output window shows the result of the execution:

```
guyashkenazi
=== Code Execution Successful ===
```

בתמונה מעלה אנו רואים את חלון הקומפילר
למדנו :

א. שכדי שהמחשב יבין מה אנו רוצים - חייבים לרשום את הפקודות בלי
שגיאות אחרת נקבל שגיעה .

ב. המחשב מאד לא סלחני לגבי שגיאות /אותיות בגודל שונה / חסר נקודה
פסיק וכו' .

ג. למדנו על ההוראה הראשונה שהיא פקודת ההדפסה

מבצעת	הפקודה
הדפסה של המלל שבתוך הגרשיים ויורדים שורה	Console.WriteLine ("ashkenazi");
הדפסה של המלל שבין הגרשיים בלי לרדת שורה	Console.Write ("guy");

משימה 1 עם ציון לעשות כרטיס ביקור - הכל באנגלית

=====

שם ומשפחה

טלפון

=====

רוני 100

עומר 100

לירון 100

זוהר 100

רפאל 100

טל 100

ליאור 100

משימה 2 עם ציון לעשות פרצוף

לירון 100

עומר 100

זוהר 100

טל 100

רפאל 100

משימה 3 - פרצוף עם כובע

לירון 100

עומר 100

רוני 100

זוהר 100

משימה 4 (1234567890) - עליכם להדפיס כל מספר בשורה נפרדת .

בצענו בעזרת הפקודה ח .

`Console.WriteLine ("1\n2\n3\n4\n5\n6\n7\n8\n9\n0");`**משימה 5 - מחשבון**

עליכם להדפיס

שדה מחושב	שזה שמודפס
15	the sum of 12+3 is
10	the mult of 5* 2 is

משימה לבית

עליכם לפתוח בגוגל דרייב , מחברת של גוגל דוקס , ושהיא תקרא "יסודות התיכנות בשפת סי שארפ"

שיעור 14.9.2025

היום נתחיל לעסוק וללמוד את נושא המשתנים , הגדרת טיפוסים המשתנים . וכך נתרגל את נושא המשתנים .

למדנו על כל משתנה , שיש להגדיר לו שם משמעותי , וכן להגדיר את הסוג של המשתנה , למשל

א. הגדרה של משתנה מסוג שלם (כלומר שיכיל ערכים שלמים) נגדיר שהסוג שלו הוא INTEGER (INT) , ולתוכו נוכל להכניס את הערך הקבוע שנרצה (שהוא בעל ערך מספרי) .

ב. כאשר נרצה להכיל משתנה שיכיל מחרוזת נשתמש ב STRING

בשיעור זה למדנו את הנושא של משתנים , למדנו. משתנה הוא כמו קופסה שעליה ניתן להכניס דברים מאותו הסוג , בעולם התוכנות למדנו שלכל משתנה נותנים שם וחשוב לתת לו שם משמעותי ; לדוגמא :

למשתנה שיכיל מספרים לא נקרא לו x אלא number1
למשתנה שיכיל שם פרטי נקרא לו FirestName
למשתנה שיכיל טיפ נקרא Tip וכו'

למדנו גם שלכל משתנה יש סוג של ערכים שהוא יכול לקבל
למשל למשתנה שיכיל מספרים שלמים מצהיר שהוא מסוג int
משתנה שיכיל מספרים ממשיים (לא שלמים) נגדיר כ double
משתנה שיכיל מחרוזות של תווים נגיד כ string
משתנה שיכיל תוו נגדיר מסוג char
משתנה שיכיל אמת או שקר נגדיר כ bool

אופן ההצהרה

משתנה מספר שלם ;// Int Namber1
 משתנה מחרוזת ;// String FirestName

למדנו שגם בשפת סי שארפ ישנה קונבנציה להגדרת שמות משתנים והיא
 אות גדולה בכל תחילת מילה .

למדנו גם על הערות בתוכנית:
 א. עבור הערה של שורה נשתמש בו //
 ב. עבור הערה של כמה שורות נתחיל בקטע כ /*
 ועדיין כ /*

***חובה בכל תוכנית לרשום הערות , מה התוכנית עושה , מה הקלט שלה
 ומה הפלט שלה

שיעורי בית : אין

שיעור 21.9.2025

התחלנו את השיעור עם חזרה ותרגול פקודות הדפסה (פרק 1 באתר
 שכולל הסבר ותרגילים) .
 המשכנו לביצוע מבדק ראשון של פקודות ההדפסה .

המשכנו עם פרק 2 משתנים ,

```
using System;

public class HelloWorld
{
    public static void Main(string[] args)
    {
        int num1=5;
        int num2= 10;
        Console.WriteLine("num1+num2="+ (num1+num2));

    }
}
```

בשיעור זה קבלתם משימה לעשות את תרגיל מספר 2 (הסבר על הצהרה והשמת משתנים)



לקישור לעבודה [לחצו כאן](#)

יש להגיש את העבודה אחרי החג מודפס בקבוצה שלנו בווצאפ .

אך יותר חשוב לעלות את המשימות לתוך המחברת הוירטואלית שלהם בגוגל דוקס .

בקבוצת הווצאפ אתם שולחים קישור למחברת שלכם (לצפייה בלבד) !! .
ואז כל תלמיד יכול לצפות במה שעשיתם וגם אני .

שיעור מתאריך 19.10.2025 3 שעות - אחרי החגים

היום בשיעורים עשינו חזרה על החומרים שלמדנו לפני החגים, הזכרנו מהי תוכנית, ואיך כותבים תוכנית, שוחחנו על פקודות ההדפסה עם ירידה ובלי ירידת שורה, `Console.WriteLine()`, חזרנו על נושא משתנים, מה המטרה שלהם, החשיבות בלתת להם שמות משמעותיים, ואיך מגדירים את הטיפוס / הסוג שלהם.

המשך היום עם הוראת הקלט `Console.ReadLine`, אשר קולטת נתון לתוך משתנה, ולמדנו שהמקלדת מחזירה תמיד ערך מחרוזתי.

כך שאם אני מבקש שם אין בעיה, או מחרוזת אחרת.

אך הבעיה נוצרת כאשר אני מבקש מספר כלשהו

שיעור מתאריך 22.10.2025 יום ד

בשיעור זה למדנו על המשתנה הקבוע `CONST`, וחשיבותו במערכת למדנו שהוא משתנה שניתן לאתחל אותו רק פעם אחת בתוכנית, ולאחר מכן מי שינסה לשנות את ערכו יקבל שגיאה. במשתנה זה משתמשים עבור ערכים שניתן לשנות אותם רק פעם אחת, והם נכונים לכל חלקי התוכנית, למשל מעמ, או מספר הימים בשבוע, הם ערכים שנכונים לכל שורות הקוד בתוכנית. כך שאם רוצים לשנות את הערך הקבוע ניתן לעשות זאת רק בשורת ההגדרה (פעם אחת).

	ערך	=	שם משתנה	סוג המשתנה	מילה שמורה
דוגמה	5000	=	MinShahr	INT	Const

תרגיל כיתה :

כתבו תוכנית מחשב אשר תבקש מהמשתמש את השכר היומי שלו ,
 התוכנית תחשב את השכר היומי כאשר היא כופלת אותו ב 24 (ימי
 העבודה בחודש) . ואז מחסירה את הסכום משכר המינימום שהוא קבוע
 במערכת ויעמוד על סכום של 5000 שח .

כתבו את התוכנית כך שתעשה זאת עבור 3 עובדים
 כלומר לכל עובד יתבקש להכניס את עלות יום העבודה , יחושב השכר
 החודשי (כפול 24) , ויופחת שכר המינימום . ויודפס השכר שלו לפני
 ההפחת מהמינימום ואחרי ההפחתה מהמינימום .

רמז : ימי השבוע צריכים להיות לא קבוע 24 אלא משתנה
 והשכר המינימום צריך להיות משתנה קבוע $CONST = 5000$.

שיעור מתאריך 16.11.2025

חזרה על קלט/פלט /משתנים

אנו בעזרת קובץ תרגילים שגיא הביא פותרים אותם למחברת .

בשלב מתקדם נכתוב אותם למחשב .

הדגשים : שמות משתנים משמעותיים

הודעות קלט לפני הקלט (הדפסות למסך)

להוסיף הערות .

תרגיל כיתה - חישוב שטח של מלבן

שטח מלבן מחושב לפי אורך כפול רוחב .

כתבו תוכנית מחשב אשר

מקבלת את אורך המלבן

ורוחב המלבן

ומחשבת את שטח המלבן ומדפיסה אותו .

תאריך 31.12.2026

מבדק מסכם

נוצר בעזרת בינה מלאכותית גמיני

והנה הפרומפט

בנה לי משחק טרוויה עם 20 שאלות בשפת סי שארפ על הנושאים הבאים , תנאים IF ELSE ,
הוראות קלט ופלט , משתנים , קבועים , לולאת FOR , CONST , מנה ושארית (MOD ׀ DIV) ,
תנאים מורכבים AND , OR NOT .

[לחצו כאן](#)



```
1: using System;
2:
3: class Program
4: {
5:     static void Main()
6:     {
7:         int total = 0;
8:         int max = 5;
9:
10:        for (int i = 1; i <= max; i++)
11:        {
12:            if (i % 2 == 0)
13:            {
14:                total = total + (i * 2);
15:            }
16:            else
17:            {
18:                total = total + i;
19:            }
20:        }
21:
22:        Console.WriteLine(total);
23:    }
24: }
```

השלימו את חומרי הלימוד שלמדנו בשיעורים הקודמים ולא סיכמנו כאן
במחברת הזו , (לפי האתר ONIM)

חומר שנלמד השבוע

תלמידים יקרים בחלק מהנשואים הבאים כבר למדנו כמו IF ELSE ,
ואופרטורים לוגיים . וגם בלולאת FOR

בשבוע זה נלמד גם

MATH
RANDOM

נחזור לתרגל ביטוי בולאיני מורכב אופרטורים
ונחזור על לולאות FOR בעיות מניה צובר וכופל .

ביום ייתנתנו לכם חוברת תרגול שבה תדרשו לעשות בבית כהכנת שיעורי
בית להגשה .

MATH
RANDOM
ביצוע מוטתנה IF .ELSE
אופרטורים לוגיים - סימני יחס
==
!=
>
<
=>
=<
ביטוי בולאיני מורכב עם אופרטורים
Not
|| OR
&& AND
לולאות FOR בעיות מניה , צובר , כופל

חלק 1: עבודה עם מחלקת Math

מחלקת **Math** היא ספרייה מוכנה ב-C# המאפשרת לבצע פעולות מתמטיות מורכבות בקלות.

פונקציות נפוצות:

- **Math.Pow(base, exponent)** – למשל 3^2 (העלאה בחזקה).
- **Math.Sqrt(num)** – שורש ריבועי.
- **Math.Abs(num)** – הופך מספר שלילי לחיובי.
- **Math.Max(a, b)** / **Math.Min(a, b)** – מציאת המקסימום או המינימום בין שני מספרים.
- **Math.Round(num)** – עיגול מספר למספר השלם הקרוב ביותר.

דוגמה:

```
double power = Math.Pow(2, 3); // תוצאה: 8
double root = Math.Sqrt(16); // תוצאה: 4
int absolute = Math.Abs(-10); // תוצאה: 10
int x = 12.4; // x=12
int y = 12.8; // y=12
int z = Math.Round(12.8); // 13
```

```
using System;

namespace MathExample
{
    class Program
    {
        static void Main(string[] args)
        {
            // 1. העלאה בחזקה (Power)
            // נחשב 2 בחזקת 3
            double baseNum = 2;
            double exponent = 3;
            double powResult = Math.Pow(baseNum, exponent);

            // 2. שורש ריבועי (Square Root)
            // נחשב שורש של 16
            double numForSqrt = 16;
            double sqrtResult = Math.Sqrt(numForSqrt);

            // 3. ערך מוחלט (Absolute Value)
            // נהפוך מספר שלילי לחיובי
            double negativeNum = -25.5;
            double absResult = Math.Abs(negativeNum);

            // הדפסת התוצאות למסך
            Console.WriteLine("--- תוצאות חישוב ---");
            Console.WriteLine($"בחזקת {exponent} של {baseNum} שווה ל: {powResult}");
            Console.WriteLine($"השורש של {numForSqrt} הוא: {sqrtResult}");
            Console.WriteLine($"הערך המוחלט של {negativeNum} הוא: {absResult}");
        }
    }
}
```

```
{absResult}");

    // מניעת סגירת החלון מיד
    Console.WriteLine("לחץ על מקש כלשהו ליציאה");
    Console.ReadKey();
}
}
}
```

תרגיל ביחד

תרגיל 1: מחשבון הנדסי בסיסי

המטרה: תרגול קליטת נתונים מהמשתמש וביצוע פעולות מתמטיות פשוטות.

תיאור התרגיל: כתוב תוכנית המבקשת מהמשתמש להזין שני מספרים:

1. מספר ראשון שישמש כבסיס (**base**).
2. מספר שני שישמש כמעריך (**exponent**).

התוכנית צריכה להציג למסך:

- את התוצאה של המספר הראשון בחזקת המספר השני.
- את השורש הריבועי של המספר הראשון (הבסיס).
- אם המשתמש הזין מספר שלילי עבור הבסיס, התוכנית תדפיס את הערך המוחלט שלו לפני ביצוע החישובים.

תרגיל 2: חישוב מרחק ושימוש בערכים מוחלטים

המטרה: יישום של מספר פונקציות מתמטיות בתוך לוגיקה אחת.

תיאור התרגיל: כתוב תוכנית שמדמה בדיקת "מרחק בקו ישר" בתוך ציר מספרים.

1. בקש מהמשתמש להזין מספר שלם כלשהו (חיובי או שלילי).

2. חשב את הערך המוחלט של המספר כדי למצוא את המרחק שלו מהנקודה 0.
3. כעת, העלה את המרחק שמצאת בחזקת 2.
4. לבסוף, הוצא שורש ריבועי מהתוצאה שקיבלת בשלב הקודם והדפס אותה.

חלק 2: מחלקת Random (הגרלת מספרים)

כדי לייצר מספרים אקראיים, עלינו ליצור "אובייקט" מסוג `Random`.

איך עושים זאת?

1. יוצרים את המגריל (פעם אחת בתחילת התוכנית).
2. משתמשים בפקודה `Next` כדי לקבל מספר.

דוגמה:

```
Random rnd = new Random(); // יצירת האובייקט
int diceRoll = rnd.Next(1, 7); // (המספר השני הוא גבול 6-ל-1 בין
(עליון לא כולל)
```

דוגמה לתרגיל מלא

```
using System;

namespace RandomExample
{
    class Program
    {
        static void Main(string[] args)
        {
            // יצירת מופע חדש של מחלקת הרנדום
            Random dice = new Random();

            // הגרלת מספר שלם בין 1 ל-10 (כולל 1, לא כולל 11)
            int targetNumber = dice.Next(1, 11);
            Console.WriteLine("המחשב הגריל מספר בין 1 ל-10. נסה!");
            Console.WriteLine("לנחש מהו");

            // קליטת ניחוש מהמשתמש
            Console.Write("הכנס את הניחוש שלך: ");
            int userGuess = int.Parse(Console.ReadLine());

            // בדיקה האם המשתמש צדק
            if (userGuess == targetNumber)
            {
                Console.WriteLine("ניחשת נכון!");
            }
            else
            {
                // כדי להראות למשתמש כמה הוא היה קרוב Math.Abs-שימוש ב
                int difference = Math.Abs(targetNumber - userGuess);
            }
        }
    }
}
```

```

        Console.WriteLine($"פספסת! המספר היה {targetNumber}.");
        Console.WriteLine($"מספרים {difference} היית רחוק רק");
        Console.WriteLine("מהתשובה הנכונה.");
    }

    Console.WriteLine("לחץ על מקש כלשהו ליציאה\");
    Console.ReadKey();
}
}
}

```

תרגיל כיתה

כתוב תוכנית מחשב אשר המחשב מגריל מספר בין 1 לבין 100 על השחקן לנחש את המספר שהמחשב הגריל בתוך 10 ניחושים מקסימום . אם המשתמש ניחש מספר גדול מהמחשב - תתקבל הודעה גדול אם המשתמש ניחש מספר קטן מהמחשב - תתקבל הודעה קטן אחרת - תתקבל הודעה כל הכבוד ניחשת בהצלחה .

בתום 10 ניחושים במידה והשחקן לא ניחש תתקבל הודעה הפסדת .

שכלול : הדפס את מספר הניחושים שבהם הצטרף המשתמש עד שהצליח (כאשר זה בתוך 10 הניחושים) . ב

חלק 3: אופרטורים של יחס (Relational Operators)

אלו הסימנים שבאמצעותם אנו משווים בין ערכים. התוצאה של השוואה כזו היא תמיד "אמת" (**true**) או "שקר" (**false**).

דוגמה	משמעות	סימן
$5 == 5$ (True)	שווה ל...	<code>==</code>
$3 != 5$ (True)	לא שווה ל...	<code>!=</code>
$20 < 10$ (False)	קטן מ...	<code><</code>
$7 > 3$ (True)	גדול מ...	<code>></code>
$10 \leq 10$ (True)	גדול או שווה	<code>=<</code>
$4 \geq 5$ (False)	קטן או שווה	<code>=></code>

חלק 4: ביצוע מותנה – מבנה If...Else

משפט ה-**if** מאפשר לתוכנית לקבל החלטות.

מבנה הקוד:

```
int age = 18;

if (age >= 18)
{
    Console.WriteLine("אתה רשאי להוציא רישיון");
}
else
{
    Console.WriteLine("צעיר מדי");
}
```

והנה IF עם שלושה מקרים (ELSE IF)

```
// checking
if (numGuess == luckNumber)
{
    Console.WriteLine("you win");
}
else if (numGuess < luckNumber)
{
    Console.WriteLine("your number is samelr then me ");
}
else
{
    Console.WriteLine("your number is bigger then me ");
}
```

כתבי תוכנית מחשב אשר מקבלת מהמשתמש שני מספרים

התוכנית בודקת אם המספר הראשון שונה מהמספר השני היא תדפיס הם שונים , אחרת היא תדפיס הם דומים .

תוכנית

חלק 5: אופרטורים לוגיים (ביטוי בולאני מורכב)

לפעמים נרצה לבדוק כמה תנאים בבת אחת.

1. **AND (&&):** מחזיר אמת רק אם כל התנאים מתקיימים.
2. **OR (||):** מחזיר אמת אם לפחות אחד מהתנאים מתקיים.
3. **NOT (!):** הופך את התוצאה (מאמת לשקר ולהיפך).

דוגמה:

```
int grade = 85;
int attendance = 90;

// בדיקה אם הציון מעל 80 וגם הנוכחות מעל 50
if (grade > 80 && attendance > 50)
{
    Console.WriteLine("סטודנט מצטיין");
}
```

למבדק שלי על נושאי הלימוד שלנו [לחצי כאן](#)



חלק 6: לולאות FOR ובעיות חישוב

לולאת **for** משמשת אותנו כאשר אנו יודעים מראש כמה פעמים נרצה לחזור על פעולה.

1. בעיית מנייה (Counter)

משמש לספירת אירועים (כמה פעמים קרה משהו).

```
int count = 0;
for (int i = 1; i <= 10; i++)
{
    if (i % 2 == 0) // אם המספר זוגי
        count++;
}
// יהיה 5 (יש 5 מספרים זוגיים בין 1 ל-10: תוצאה)
```

2. בעיית צובר (Accumulator - סכום)

משמש לחיבור ערכים מצטברים.

```
int sum = 0;
for (int i = 1; i <= 5; i++)
{
    sum += i; // לסכום הקיים i מוסיף את
}
// תוצאה: 15 = 1+2+3+4+5
```

3. בעיית כופל (Product)

משמש למציאת מכפלה של סדרת מספרים (חשוב להתחיל ב-1!).

1. המחלקה Math (ארגז הכלים המתמטי)

תחשבו על **Math** כמו על מחשבון מדעי שמותקן בתוך השפה. היא מכילה המון פקודות מוכנות לחישובים.

חשוב לדעת: ב-**Math** אנחנו לא צריכים ליצור שום דבר חדש, פשוט כותבים את השם שלה ובוחרים פעולה.

פקודות נפוצות:

- **Math.Max(a, b)**: מחזירה את המספר הגדול מבין השניים.
- **Math.Min(a, b)**: מחזירה את המספר הקטן מבין השניים.
- **Math.Pow(base, exponent)**: חזקה. לדוגמה
Math.Pow(2, 3) ייתן 8 (2 בחזקת 3).
- **Math.Sqrt(num)**: שורש ריבועי. לדוגמה
Math.Sqrt(16) ייתן 4.
- **Math.Abs(num)**: ערך מוחלט (הופך מספר שלילי לחיובי).
- **Math.Round(num)**: עיגול מספרים למספר השלם הקרוב ביותר.

דוגמה לקוד:

```
double powerResult = Math.Pow(5, 2); // 25 תוצאה:
double squareRoot = Math.Sqrt(49); // 7 תוצאה:
int biggerNumber = Math.Max(10, 20); // 20 תוצאה:

Console.WriteLine(biggerNumber);
```

2. המחלקה Random (הקוביה של המחשב)

אם **Math** היא המחשבון, אז **Random** היא הקובייה. אנחנו משתמשים בה כשאנחנו רוצים להגריל מספרים (למשל: במשחקים, בחירת משתמש אקראי וכו').

שימו לב: בניגוד ל-Math, כאן אנחנו **חייבים** ליצור "מייצר מספרים" חדש לפני שמשתמשים בו.

איך משתמשים בזה?

1. יוצרים את המשתנה: `Random rnd = new Random();`
2. מבקשים מספר: `int num = rnd.Next(1, 11);`

חוק ה-Next החשוב:

היי! איזה כיף שאתה ממשיך ללמוד ולפתח את היכולות שלך ב-C#.

כשמתחילים לתכנת, מהר מאוד מגלים שלא תמיד אנחנו רוצים לעשות רק פעולות חשבון פשוטות כמו חיבור (+) או חיסור (-). לפעמים אנחנו צריכים שהמחשב יהיה "חכם" יותר (יעשה חישובים מסובכים) או "מפתיע" יותר (יבחר מספרים בצורה אקראית).

בשביל זה קיימות שתי מחלקות (Classes) מרכזיות: **Math** ו-**Random**.

1. המחלקה Math (ארגז הכלים המתמטי)

תחשבו על **Math** כמו על מחשבון מדעי שמותקן בתוך השפה. היא מכילה המון פקודות מוכנות לחישובים.

חשוב לדעת: ב-Math אנחנו לא צריכים ליצור שום דבר חדש, פשוט כותבים את השם שלה ובוחרים פעולה.

פקודות נפוצות:

- **Math.Max(a, b):** מחזירה את המספר הגדול מבין השניים.
- **Math.Min(a, b):** מחזירה את המספר הקטן מבין השניים.
- **Math.Pow(base, exponent):** חזקה. לדוגמה `Math.Pow(2, 3)` ייתן 8 (2 בחזקת 3).
- **Math.Sqrt(num):** שורש ריבועי. לדוגמה `Math.Sqrt(16)` ייתן 4.
- **Math.Abs(num):** ערך מוחלט (הופך מספר שלילי לחיובי).
- **Math.Round(num):** עיגול מספרים למספר השלם הקרוב ביותר.

דוגמה לקוד:

```
#C
double powerResult = Math.Pow(5, 2); // תוצאה: 25
double squareRoot = Math.Sqrt(49); // תוצאה: 7
```

```
(int biggerNumber = Math.Max(10, 20 // תוצאה: 20
```

```
);(Console.WriteLine(biggerNumber
```

2. המחלקה Random (הקוביה של המחשב)

אם **Math** היא המחשבון, אז **Random** היא הקובייה. אנחנו משתמשים בה כשאנחנו רוצים להגריל מספרים (למשל: במשחקים, בחירת משתמש אקראי וכו').

שימו לב: בניגוד ל-Math, כאן אנחנו **חייבים** ליצור "מייצר מספרים" חדש לפני שמשתמשים בו.

איך משתמשים בזה?

1. יוצרים את המשתנה: `Random rnd = new Random();`
2. מבקשים מספר: `(int num = rnd.Next(1, 11);`

חוק ה-Next החשוב:

כשכותבים `Next(min, max)`, המחשב כולל את המספר הראשון (המינימום) אבל **לא כולל** את המספר האחרון (המקסימום).

פקודה	מה היא עושה?
<code>rnd.Next()</code>	(מחזירה מספר שלם אקראי כלשהו (חיובי).
<code>rnd.Next(10)</code>	(מחזירה מספר בין 0 ל-9 (לא כולל 10).
<code>rnd.Next(1, 7)</code>	(מחזירה מספר בין 1 ל-6 (כמו הטלת קובייה).
<code>rnd.Next(6)+5</code>	מחזירה מספרים בין 5 עד 10 (כלומר מגרילה מספרים מ 0 עד 5 כולל) ולכל מספר שהיא מגרילה היא מוסיפה 5 .

```
Random dice = new Random(); // יצירת האובייקט
int roll = dice.Next(1, 7); // הגרלת מספר בין 1 ל-6
Console.WriteLine("המספר שיצא בקובייה הוא " + roll);
```

תרגילי סיכום - להגשה והצגה בכיתה .

1. תרגיל Math ו-Random: הגרל שני מספרים בין 1 ל-100. הדפס את הגבוה מביניהם ואת השורש הריבועי שלו (מעוגל).
2. תרגיל לוגיקה: קלוט מהמשתמש גיל וציון פסיכומטרי. הדפס "התקבלת" אם הגיל מעל 20 או שהציון מעל 700.
3. תרגיל לולאות: כתוב לולאה שקולטת 10 מספרים מהמשתמש. בסוף הלולאה הדפס:
 - כמה מספרים היו חיוביים (מונה).
 - מה סכום כל המספרים שנקלטו (צובר).
4. כתוב תוכנית מחשב אשר מגרילה 2 קוביות שש בש , התוכנית תדפיס את שתי הקוביות .
- בנוסף תבדוק אם יצא דאבל (שני המספרים על הקוביות אותו הדבר) , אז המחשב ידפיס יצא דאבל .
5. כתוב תוכנית המגרילה 10 פעמים קובייה , ואז התוכנית תדפיס כמה פעמים יצא מספר זוגי , וכמה פעמים יצא המספר 6 .

האופרטורים הלוגיים המרכזיים

כדי ליצור תנאי מורכב, נשתמש בשלושת הכלים הבאים:

דוגמאות קוד

1. שימוש ב-"וגם" (&&)

נשתמש בזה כשנרצה לוודא שטווח מסוים מתקיים. למשל, בדיקה אם גיל מתאים לנער (בין 13 ל-19):

```
int age = 15;

if (age >= 13 && age <= 19)
{
    Console.WriteLine("אתה נער.");
}
```

2. שימוש ב-"או" (||)

נשתמש בזה כשיש כמה אפשרויות שונות שמאשרות את התנאי. למשל, בדיקה אם היום הוא יום מנוחה:

```
string day = "Saturday";

if (day == "Friday" || day == "Saturday")
{
    Console.WriteLine("סוף שבוע! אפשר לנוח.");
}
```

3. שילוב של תנאים (סוגריים)

כמו במתמטיקה, סוגריים קובעים מה יחושב קודם. למשל, כניסה למועדון צריך להיות מעל גיל 18 וגם שיהיה לך כרטיס או שתהיה ברשימת ה-VIP):

```
int age = 20;
bool hasTicket = false;
bool isVip = true;

if (age >= 18 && (hasTicket || isVip))
{
    Console.WriteLine("ברוך הבא למועדון");
}
```

```
}
```

שאלות הבנה לתלמיד

נסה לענות על השאלות הבאות כדי לוודא שהבנת את הנושא:

שאלה 1: מה תהיה התוצאה? נתון הקוד הבא:

#C

```
int x = 10;
int y = 5;
if (x > 8 && y < 3) {
    Console.WriteLine("אמת");
} else {
    Console.WriteLine("שקר");
}
```

מה יודפס למסך? הסבר למה.

שאלה 2: כתיבת תנאי כתוב שורת `if` שבודקת האם מספר `num` הוא גם חיובי (גדול מ-0) וגם זוגי (רמז: השתמש ב-`num % 2 == 0`).

שאלה 3: ניתוח לוגי נתון המשתנה `isRaining = true`. מה הערך של הביטוי הבא: `!(isRaining)`?

שאלה 4: תרגיל מאתגר מערכת כניסה לאתר דורשת שם משתמש וסיסמה. כתוב קוד הקולט מהמשתמש שני מחרוזות (`username` ו-`password`). התוכנית תדפיס "כניסה הצליחה" רק אם שם המשתמש הוא "admin" והסיסמה היא "1234". אם אחד מהם שגוי, הדפס "פרטים שגויים".

סיכום של החומר שלמדנו ביחד

נושא	הסבר קצר	ערך התחלתי נפוץ
מנייה (Counter) $x=x+1$; $x++$; $x--$; $x-=1$	ספירת כמות הפעמים שאירוע מסוים קרה (למשל: כמה תלמידים עברו את המבחן) ²²² .	0
צבירה (Accumulator) $y=y+m$; $y+=m$; $y-=m$; $y-=m$	חישוב סכום של ערכים (למשל: מה סכום הציונים של כל הכיתה) ⁴⁴⁴⁴ .	0
כפל (Multiplier) $z=z*m$; $z*=m$; $z/=m$; $z/=m$	הכפלה חוזרת של ערכים (למשל: חישוב עצרת או חזקה) ⁶⁶⁶⁶⁶ .	1
מינימום / מקסימום $\min(1, 6, 7, \underline{0}, 2)$ $\max(12, 2, 34, 1, 45)$	מציאת הערך הכי גבוה או הכי נמוך בסדרה של נתונים ⁸ .	האיבר הראשון בסדרה
אחרון ולפני אחרון	שמירת מספר ערכים בולטים (כמו מקום)	ערך קיצוני (כמו double.MaxValue) ¹¹

	ראשון, שני ושלישי) 10101010.	
1 ¹³ (עבור רצף נוכחי)	מעקב אחר רצף של נתונים (למשל: כמה ימים רצופים הייתה עלייה בטמפרטורה) 12121212. 1+	רצפים (Sequences)

הסבר על רצפים בעיות מקסימום ומינימום

בסדרה של מספרים (כאשר נתון שהסדרה **לא ממויינת** / לא לפי סדר עולה או יורד) ,
אנו נדרשים כדי למצוא את המספר המקסמלי או המינמלי לעבור על כל ערכי הסדרה
ולבדוק מולם .

אופן הבדיקה מקסימום :

א. לוקחים את האיבר הראשון בסדרה לתוך משתנה MAX , וקבועים שהוא המספר
המקסמלי , לאחר מכן מחפשים בסדרת המספרים מישוה גדול ממנו , אם מצאנו כזה
אז אנו מחלפים את הערך הישן בערך החדש הגדול יותר . בסוף הפעולה אנו נשארים
עם הערך המקסמלי .

ב. אופן הבדיקה מינימום :

אותו הדבר כמו ברצף של מספרים שמחפשים את המקסימום , רק שכאן לתוך
המשתנה MIN , מכניסים את הערך הראשון , וכעת עוברים על כל הרצף ומחפשים מי
יותר קטן ממנו , ומי שכן יותר קטן ממנו יחליף את הערך בערך החדש .

בסדרת מספרים ממויינת (כלומר היא לפי הסדר עולה או יורד)

כדי למצוא את המקסימום אז בסדר עולה , הוא תמיד יהיה בסוף , ובסדרה יורדת הוא
תמיד יהיה בהתחלה .

וכדי למצוא את המינמום אז בסדר עולה הוא ייהיה בהתחלה , ובסדר יורד הוא יהיה בסוף .

תרגיל בית :
כתבו תוכנית אשר קולטת מהמשתמש 10 מספרים שונים , ועליכם להדפיס בסוף התוכנית מי המספר הגדול ביותר (מקסימום) ומי הקטן ביותר .

מבדק כיתה על הטבלה למעלה [לחצו כאן](#)



פירוט הנושאים

1. בעיות מנייה (Counters)

משתמשים במונה כאשר רוצים לדעת כמה פעמים תנאי מסוים התקיים.

- איך זה עובד? מגדירים משתנה (למשל `count`), מאתחלים ל-0, ובכל פעם שהתנאי מתקיים בתוך הלולאה, מגדילים אותו ב-1 (`++count`).
- דוגמה: ספירת כמה מושבעים הצביעו "אשם" (y) מתוך 12 חברים.
1+

2. בעיות צובר וכופל (Accumulator & Multiplier)

- צובר (Sum): משמש לסיכום ערכים. אנחנו מוסיפים למשתנה את הערך החדש שנקלט: `sum = sum + num`.
+1
- כופל (Mult): משמש למכפלה. חשוב לאתחל ל-1 (כי כפל ב-0 תמיד ייתן 0):
`mult = mult * i`.
+2
- שימוש נפוץ: חישוב ממוצע (סכום חלקי כמות) או חישוב עצרת (! n).
+3

3. מציאת מינימום ומקסימום

כדי למצוא את הערך הכי גדול (Max), אנחנו משווים כל מספר חדש ל"שיא" שמצאנו עד כה.

1+

- השיטה המומלצת: קולטים את המספר הראשון ומגדירים אותו כ-Max ההתחלתי. לאחר מכן, בלולאה שרצה על שאר המספרים, אם מצאנו מספר

גדול מה-Max, הוא הופך להיות ה-Max החדש.
1+

4. שמירת מספר ערכים (אחרון ולפני אחרון)

לפעמים צריך לשמור לא רק את המקום הראשון, אלא גם את השני והשלישי.
1+

- **הטכניקה:** כשמוצאים "שיא" חדש למקום הראשון, מעבירים את הערך הישן של המקום הראשון למקום השני, ואת השני לשלישי, ורק אז מעדכנים את הראשון. זה נקרא "הזזה" של ערכים.

5. לולאות מקוננות (Nested Loops)

זוהי לולאה שנמצאת בתוך לולאה אחרת.

- **דוגמה קלאסית:** שעון (סטופר). הלולאה החיצונית סופרת שעות, בתוכה לולאה לדקות, ובתוכה לולאה לשניות. בכל פעם שהלולאה הפנימית מסיימת סיבוב מלא, הלולאה שמעליה מתקדמת ב-1.
3+

תרגילי בית כהחלפה למבחן

יסודות שפת C#: קלט/פלט, משתנים, לולאות for, צבורים (סכום וממוצע), מונים ותנאים מורכבים.

שיעורי הבית בתאריך 14.1.2026

עמוד 40 תרגיל 1 : אלוף הגיימינג - חלק א

חוברת תרגול: יסודות התכנות ב-C# (כהכנה לבחינה)

תרגיל 1: אלוף הגיימינג - חלק א'

במשחק מחשב, שחקן עובר מספר שלבים. בכל שלב הוא מקבל ניקוד (מספר שלם בין 0 ל-1,000).

שחקן נחשב "שחקן מצטיין" אם ממוצע הניקוד שלו בכל השלבים גבוה מ-750.

דוגמה:

שחקן שיחק 3 שלבים וקיבל: 900, 600, 800.

הממוצע הוא: $766.66 = \$ (800+600+900) / 3$ ולכן הוא שחקן מצטיין.

המשימה:

כתוב תוכנית ב-C# הקולטת מהמשתמש את מספר השלבים ששיחק, ולאחר מכן קולטת את הניקוד של כל שלב. התוכנית תדפיס האם השחקן מצטיין או לא.

תרגיל 2: אלוף הגיימינג - חלק ב' (תנאים מורכבים)

באותו משחק, הקריטריונים להצטיינות הפכו למורכבים יותר. שחקן נחשב "שחקן מצטיין" אם מתקיים לפחות אחד מהתנאים הבאים:

1. ממוצע הניקוד בכל השלבים גבוה מ-750.

2. לפחות 50% מהשלבים הסתיימו בציון מעל 850.
3. יש לפחות שלב אחד שבו השחקן קיבל ניקוד מושלם של 1,000.

המשימה:

1. כתוב תוכנית ב-C# הקולטת את מספר השלבים ואת הציונים, ומדפיסה האם השחקן מצטיין לפי התנאים החדשים.
2. טבלת מעקב: בצע טבלת מעקב עבור הקלט הבא: 4 שלבים, ציונים: 500, 1000, 400, 600.
- (הטבלה תכלול: עמודה לכל משתנה, עמודה לכל תנאי בוליאני ועמודת פלט).

תרגיל 3: אתגר הצעדים (כושר גופני)

- אפליקציית כושר עוקבת אחרי מספר הצעדים של משתמש במשך שבוע.
- משתמש נחשב "פעיל" אם מתקיים לפחות אחד מהתנאים הבאים:
1. ממוצע הצעדים היומי שלו גבוה מ-8,000.
 2. לפחות ביומיים מתוך השבוע הוא עבר את ה-10,000 צעדים.
 3. היה לפחות יום אחד שבו הוא הגיע ל-15,000 צעדים.

המשימה:

1. כתוב תוכנית ב-C# הקולטת מהמשתמש את מספר הצעדים שביצע בכל אחד מ-7 ימי השבוע ומדפיסה האם הוא "משתמש פעיל".
2. **טבלת מעקב:** בצע טבלת מעקב עבור הקלט הבא (7 ימים): 5000, 11000, 4000, 11000, 3000, 2000, 6000.

תרגיל 4: חממה חכמה (עבודה עם מספרים אקראיים)

במערכת חקלאית, חיישן מודד טמפרטורה במספר ערוגות (בין 5 ל-15 מעלות צלזיוס).

חממה נחשבת "מאוזנת" אם מתקיים לפחות אחד מהתנאים הבאים:

1. ממוצע הטמפרטורה בכל הערוגות הוא בין 10 ל-12 מעלות (כולל).
2. לפחות 70% מהערוגות הן בטמפרטורה של מעל 13 מעלות.
3. יש לפחות ערוגה אחת שהטמפרטורה בה היא בדיוק 15 מעלות.

המשימה:

1. כתוב תוכנית ב-C# המגרילה מספר שלם המייצג כמה ערוגות יש בחממה (מספר בין 3 ל-6).
2. עבור כל ערוגה, התוכנית תגריל טמפרטורה (מספר שלם בין 5 ל-15).
3. התוכנית תדפיס את מספר הערוגות, את הטמפרטורה של כל ערוגה, ואת ההודעה האם החממה "מאוזנת" לפי התנאים.

דגשים לפתרון (טיפים לתלמיד):

- **ממוצע:** זכור להשתמש במשתנה מסוג `double` כדי לקבל תוצאה מדויקת בחילוק.
- **אחוזים:** כדי לחשב אם 60% מהמקרים קרו, השתמש במונה (`counter`) ובדוק בסוף הלולאה האם: `(double)count / total >= 0.6`.
- **דגל (Flag) או מונה:** למציאת "לפחות אחד ששווה ל-100", ניתן להשתמש במונה שבודק אם הערך גדול מ-0 בסוף, או במשתנה בוליאני שמתעדכן ל-`true`.

שאלה מספר 8

השאלה היא מה שלומך

פתרון התרגיל מספר 8 שלי

// See <https://aka.ms/new-console-template> for more information

```
for (int i = 1; i < 5; i++)
{
    Console.WriteLine(i);
    for (int j = 1; j <= 5; j++)
    {
```

```
    Console.Write(j);  
    }  
    Console.WriteLine("");  
}
```

ההרצה

שיעור מ-11.3.2026 - פעולות

להלן פתרון תרגיל הבית מספר 4 (במשימה של פעולה מספר 11 באתר
(ONIM

```
//=====
=====
//
public static void SumDigit(int number)

{
    // זו הפעולה שמקבלת מספר וסוכמת את סכום הספרות שלו
    //

    int digit3 = number % 10; //7
    int digit2 = (number / 10) % 10; //7 המטרה להגיע למספר
    האמצעי

    int digit1 = (number / 100); //7

    Console.WriteLine("the sum of {0} is {1}", number,
    (digit1+digit2+digit3));

}

//=====
=====

static void Main(string[] args)
{
```

```

int num;
int[] arrNum= new int[10];//תאים 10
for (int i=0;i<10;i++)
{
    Console.WriteLine("insert num 3 digit number{0}",i);
    num = int.Parse(Console.ReadLine());
    arrNum[i] = num;//המשתמש
    בחר לתוך התאים במערך
    SumDigit(num);//המספר שקבלתי
    מהמשתמש
}
}
}
}

```

שיעורי בית

1. כתבו תוכנית מחשב המקבלת מספר דו ספרתי
א. על התוכנית לבצע בפעולות שונות , חיבור , חיסור , כפל וחילוק

למשל פעולה של כפל אשר תדפיס את הספרה הראשונה כפול השנייה
למשל פעולה חיבור אשר תדפיס את הספרה הראשונה ועוד השנייה
למשל פעולה חיסור אשר תדפיס את הספרה הראשונה פחות השנייה
ולמשל חילוק אשר תדפיס את התוצאה (המלאה) של החילוק של הספרה
הראשונה בספרה השנייה .

לדוגמה המשתמש קלט את המספר 23

התוכנית תדפיס

הכפל הוא 2 כפול 3 שזה 6

החיבור הוא 2 ועוד 3 שזה 5

החיסור הוא 2 פחות 3 שזה -1

והחילוק הוא 2 לחלק 3 שזה התוצאת חילוק

2. לחברת כח אדם ביקשה מעובדיה לבחור קוד סודי , הקוד
א. חייב להיות בן מספר תלת ספרתי שכל הספרות שלו חיבות להיות
שונות , אם המספר שונה אז המערכת תדפיס כל הכבוד זה מספר
סודי טוב . אחרת המספר לא חוקי . (נא להשתמש בפעולות) .

שיעור מקוון 15.3.2026

הנושא פעולות מחזירות ערך
 בשיעור זה היום נלמד ונכיר פעולות ש :
 א. לא מקבלות פרמטרים אך מחזירות ערך
 ב. כן מקבלות פרמטרים וכן מחזירות ערך

מבנה הפעולה שלא מקבלת פרמטרים וכן מחזירה ערך

```
//=====
//פעולה שלא מקבלת כלום אך מחזירה תשובה
public static [סוג הערך המחזור] PEOLANAME()
{
    כאן ההוראות שאנו רוצים שהפעולה תבצע
    return 5;// כאן הערך שהפעולה מחזירה
}
//=====
```

פעולה יכולה להחזיר כל סוג של משתנים שלמדנו ועוד .

הנה דוגמה לפעולה שמחזירה מספר שלם

```
//=====
//פעולה שלא מקבלת כלום אך מחזירה תשובה
public static int retrun5()
{
    Console.WriteLine("lo kiblati klom");
    return 5;
}
//=====
```

הפעולה למעלה לא מקבלת כלום אך מחזירה ערך מספר שלם , במקרה שלה היא מחזירה 5 .

אופן הקריאה לפעולה בתוכנית הראשית MAIN

```
int number = retrun5();
```

הנה דוגמה לפעולה שלא מקבלת כלום אך מחזירה מחרוזת

```
//=====
public static string printname()
{
    return "guy";
}
//=====
```

הפעולה הזו לא מקבלת כלום ומחזירה את המחרוזת guy

אופן הקריאה לפעולה

```
string name = printname();
Console.WriteLine(name);
```

דוגמה לפעולה שלא מקבלת כלום ומחזירה ערך בולאני

```
//=====
public static bool isBOY()
{
    return true;
}

//=====
static void Main(string[] args)
{

    //קריאה לפעולה הבולאנית
    bool isStudent = isBOY();
    Console.WriteLine("the answer is {0}", isBOY());
}
```

*הערה חשובה : כל המשתנים שאתם מגדירים בתוך הפעולה מסויימת לא מוכרים על ידי התוכנית הראשית וגם לא על ידי פעולות אחרות . אלא מוכרות רק בתוך הפעולה .

תרגיל בית

יקרים סיכום חומר הלימוד במחברת המשותפת .

שיעורי בית

עליכם לכתוב תוכנית מחשב אשר מדפיסה את הציור הבא :
מותר להשתמש רק בפעולות שלא מקבלות ערכים ולא מחזירות דבר

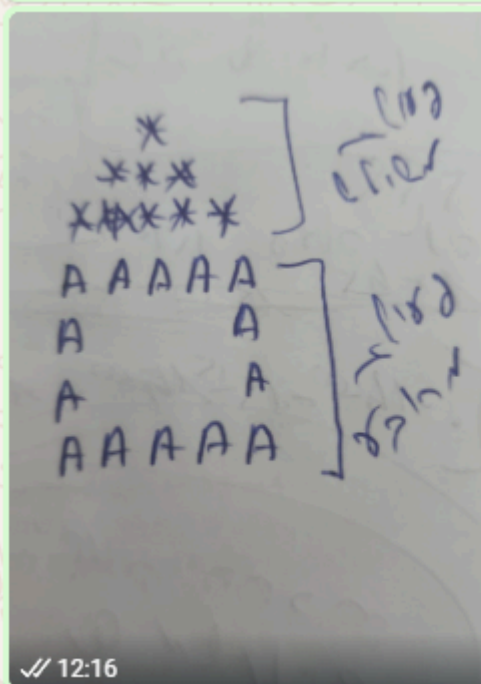
הנה הצילום למה שנדרש לראות על המסך .

א. שימו לב יש לפתור עם 2 פעולות

1. פעולה המציירת את הגג המשולש של הבית

2. פעולה המציירת את המרובע של הבית

✓ 12:16



✓ 12:16

```

;using System
;using System.Collections.Generic
;using System.Linq
;using System.Runtime.Remoting.Channels
;using System.Text
;using System.Threading.Tasks

namespace ConsoleApp5
{
    internal class Program
    {
        =====//
        ()public static void printX
        {
            ;("Console.WriteLine("x
            {

            =====//

            //כאן יצרתי פעולה שמקבלת שלושה ערכים ומדפיסה אותם
            (public static void printMalben(int orch, int rochav, string color
            {
                ;(Console.WriteLine("ORCH ={0}", orch
                ;(Console.WriteLine("ROCHAV ={0}", rochav
                ;(Console.WriteLine("color ={0}", color

            {

            =====//
            //פעולה שלא מקבלת כלום אך מחזירה תשובה
            ()public static int retrun5
            {
                ;("Console.WriteLine("lo kiblati klom
                ;return 5
            {
            =====//
            ()public static string printname
            {
                ;"return "guy

            {
            =====//
            ()public static bool isBOY
            {
                ;return true
            {

            =====//
            //פעולה שכן מקבלת פרמטרים וכן מחזירה ערך

```

```
(public static void scom2Numbers(double num1,int num2
}
```

```
;double x = num1 + num2
;(Console.WriteLine(num1+num2 //
;(return (num1 + num2//
{
```

```
=====//
```

```
(public static int printSum(int x,int y
}
```

```
;(Console.WriteLine(x + y
```

```
{ ;return x + y
```

```
=====//
```

```
(static void Main(string[] args
}
```

קריאה לפעולה שמקבלת ערכים ומחזירה ערך ממשי

```
;(scom2Numbers(5.0, 4 //
```

```
;double x = 8//
```

```
;(Console.WriteLine(x//
```

קריאה לפעולה הבולאנית

```
;(bool isStudent = isBOY //
```

```
;(Console.WriteLine("the answer is {0}", isBOY//
```

```
;(printSum(4, 5
```

```
;(string name = printname //
```

```
;(Console.WriteLine(name //
```

```
guy ידפיס//;Console.WriteLine(printname //
```

```
;(int number = retrun5 //
```

```
;()Console.WriteLine("kibalti ={0}", retrun5//
```

```
//("printMalben(45, 30, "green//  
//();()printX//  
{  
  
{  
{
```

מפגש בגוגל מיט יום ב - 16/03/2026

היום במפגש

נשנן את הנושא של פעולות

עד כה למדנו פעולות

1. שלא מחזירה כלום ולא מקבלת כלום
2. שלא מחזירה כלום אך מקבלת פרמטרים
3. שמחזירה (תמיד ערך אחד) , ומקבלת אחד או יותר פרמטרים .

א. נכיר את המקרה שבו פעולה מחזירה ערך ואנו מתעלמים ממנה

ב. נכיר את השגיאה שפעולה VOID לא צריכה אך רושמים RETURN

ג. נתרגל תרגילי כיתה

ד. נקיים מבדק הצלחה מקוון בשעה הבאה דרך האתר .

כתבו פעולה אשר מקבלת את האורך והרוחב של מלבן , הפעולה תחשב ותחזיר לקורא את שטח המלבן (שטח המלבן = אורך כפול רוחב)

הנה הקוד שלמדנו היום של פעולה שמחשבת את השטח של משולש על ידי קליטת אורך ורוחב מהמשתמש

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Remoting.Channels;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp5
{
    internal class Program
    {
        // יש לך משולש שגובה שלו 10 סמ , ורוחב שלו 2 סמטבר מה/
        שטח המשולש
        // גובה כפול רוחב לחלק ל 2
        // הפעולה מקבלת שני נתונים של אורך ורוחב והפעם מחזירה לי/
        את השטח
        public static double Setach(int orech , int rochav)
        {
            double s = (orech * rochav) / 2;

            return s;
        }

        //=====
        static void Main(string[] args)
        {
```

```
// קולט מספר מהמשתמש עבור אורך המשולש
Console.WriteLine("inset orech ");
int num1 = int.Parse(Console.ReadLine());

// קולט מספר מהמשתמש עבור רוחב המשולש
Console.WriteLine("inset rochav ");
int num2 = int.Parse(Console.ReadLine());

// קורא לפעולה שמחשבת את האורך כפול הרוחב לחלק
// לשתיים ומחזירה את שטח המשולש המחושב
double s2 = Setach(num1, num2);
Console.WriteLine("the shetach is ={0}",s2);

    }

}
```

שיעור 18.3.2026 יסודות התיכנות - המשך פעולות

כתוב תוכנית אשר מקבלת שלושה פרמטרים

א. מספר התווים שנדרש להדפיס

ב. סוג התוו שנדרש להדפיס

ג. האם לרדת שורה בין תו לתו או לעשות אחד ליד השניה

הנה התוכנית בכיתה

```

פעולה המקבלת מספר ומדפיסה בשורה אחת כמות של
כוכביות אחת ליד השניה לפי המספר
public static void printStar(int count,char tav,bool
isEnter)

// טענת כניסה
//count=> מספר התווים שיש להדפיס על
המסך
//tav=> סוג התוו שנדרש להדפיס
// if isEnter = TRUE => הדפיס את האותיות אחת ליד השניה
// if isEnter = FALSE => הדפיס את האותיות אחת מתחת
לשניה

{
    for (int i = 0; i < count; i++)
    {
        if (isEnter == true)
            Console.Write(tav);
        else
            Console.WriteLine(tav); // מה שהפעולה מדפיסה
או מחזירה זו טענת היציאה
    }
}

```

```

        Console.WriteLine();

    }

    static void Main(string[] args)
    {
        printStar(4,'A',true);//AAAA
        printStar(2,'B',false);//BB

    }
}

```

תרגיל כיתה

קוד כניסה של עובד לחברת "Cyber Now" הינו תקין אם הוא מקיים את התנאים הבאים :

א. מתחיל בתווים CN

ב. מכיל לפחות פעמיים את אחד מהתווים המיוחדים הבאים : \$ # !

ג. מסתיים בתווים CN

ד. אורכו לפחות 8 תווים

יש לכתוב פעולות / פעולה , אשר מקבלת כפרמטר מחרוזת המייצגת קוד כניסה של עובד ומחזירה TRUE אם הקוד תקין , אחרת תחזיר FALSE .

לא חוקי כי סעיף ב לא מתקיים=>CN#123456789CN
לא חוקי משום שלא מקיים את סעיף ג =>CN#123\$3

פתרון פעולה רק לסעיף א

קוד לדוגמה (מחרוזת)

0	1	2	3	4
C	N	1	2	3

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Remoting.Channels;
using System.Security.Cryptography.X509Certificates;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp5
{
    internal class Program
    {
        //=====
        //פעולה שמקבלת קוד ובודקת אם שתי האותיות הראשונות הן
        CN
        public static bool IsFirestCN(string Code)
        {
            if ((Code[0] == 'C') && (Code[1] == 'N'))
                return true;
            else return false;
        }

        //=====
        =====
    
```

```
static void Main(string[] args)
{
    bool isTrue = IsFirestCN("CN123456");
    Console.WriteLine(isTrue);
}
}
```

פתרון סעיף ב

פעולה המקבלת מחרוזת וסורקת אותה ובודקת כמה פעמים יש תווים שהם
 או # , או \$ או ! . והיא תספור אותם ,
 ואז היא תבדוק אם המספר הזה הוא לפחות 2 אז תחזיר אמת אחרת
 תחזיר שקר .

הקוד	כמות	ערך מוחזר
CN#123	1	FALSE
CN34#@!B	2	TRUE
CN####\$D#	6	TRUE
CNFRDESDER455 565	0	FALSE

לשם הפתרון אני צריך להשתמש בפונקציה של מחרוזת שאומרת לי מה
 אורך המחרוזת

code.Length

למשל עבור ABCD <= הפונקציה תחזיר לי 4
 ועבור C <= הפונקציה תחזיר לי 3

הנה התוכנית הסופית הבודקת את תנאי א וגם תנאי ב

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Remoting.Channels;
using System.Security.Cryptography.X509Certificates;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp5
{
    internal class Program
    {
        //=====
        //=====
        // CN פעולה שמקבלת קוד ובודקת אם שתי האותיות הראשונות הן
        public static bool IsFirestCN(string Code)
        {
            if ((Code[0] == 'C') && (Code[1] == 'N'))
                return true;
            else return false;
        }

        //=====
        //=====
        // פעולה אשר מקבלת קוד ובודקת כמה פעמים מופיעים התווים
        // #!$
        // אם לפחות 2 יחזיר אמת אחרת מחזירה שקר
        public static bool Is2Digit (string code)
        // ABC
        {
            int Count = 0;

```

```

        // לולאה שעוברת על כל המחרוזת וסופרת את כמות התווים
        החייבים
        //$!
        for (int i = 0; i < code.Length; i++)
        {
            if ((code[i] == '#') || (code[i] == '$') || (code[i] == '!'))
                Count++;
        }

        // כאן אנו בודקים אם הכמות היא לפחות 2 תווים ממה שהגדרנו
        אם כן תחזיר אמת אחרת שקר
        if (Count >= 2) return true;
        else return false;

    }

//=====
=====
    static void Main(string[] args)
    {

        string code = "CN123456#df$";
        //בדיקה של תנאי הראשון
        bool isTrue1 = IsFirestCN(code);
        Console.WriteLine("chek 1 = {0}",isTrue1);

        //בדיקה של התנאי השני
        bool isTrue2 = Is2Digit(code);
        Console.WriteLine("chek 2 = {0}",isTrue2);

    }

}

```

שיעור מפגש 22.03.2026 יום ראשון שעה 10:10

היום נמשיך ונלמד את המשך התרגול של פעולות בשיעור הקודם השבנו על תרגיל כאשר חילקנו את הפתרונות לתתי תוכניות , כך שכל תת תוכנית הינה פעולה , בשיעור זה נפתור יחד את פעולה עבור הקוד המסתיים ב CN ונמשיך .

להזכירכם התרגיל היה לכתוב תוכנית שבודקת שהקוד העובד חוקי קוד חוקי הוא :

א. מתחיל ב CN

ב. מכיל לפחות פעמיים את אחד מהתווים המיוחדים הבאים \$!#

ג.הקוד מסתיים ב CN

ד. אורכו של הקוד לפחות 8 תווים .

כל התנאים חייבים להתקיים , מספיק שתנאי אחד לא מתקיים והקוד לא חוקי

בשיעור זה : ראינו למעשה גם שייתכן ותהיה בעיה למשל אם המשתמש ירשום את הקוד עם c או CN , או cN וכו כלומר שילוב של אותיות קטנות וגדולות , ולכן כאשר אנו מבצעים בדיקות לתוכנית שלנו , אנו עולים על זה ומשפרים את המערכת . כי אם היינו משווים רק בין CN והמשתמש היה כותב c הוא היה מקבל שגיאה שאין CN בהתחלה והוא עלול שלא להבין מדוע .

בנוסף הוספנו בכל פעולה שבודקת תנאי, שאם התנאי הנבדק מחזיר שקר (נכשל) לפני כן הוא רושם הערה , כך שהמשתמש ידע בדיוק היכן השגיאה (כל פעולה תדווח על החלק שלה מה השגיאה) .

הנה התוכנית הסופית שעשינו בשיעור

```
using System;
```

```

using System.Collections.Generic;
using System.Linq;
using System.Runtime.Remoting.Channels;
using System.Security.Cryptography.X509Certificates;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp5
{
    internal class Program
    {
        //=====
        //פעולה שמקבלת קוד ובודקת אם שתי האותיות הראשונות הן CN
        public static bool IsFirestCN(string Code)
        {
            if ((Code[0] == 'C') && (Code[1] == 'N'))
                return true;
            else
            {
                Console.WriteLine("ERROR the code not start with
CN");
                return false;
            }
        }
        //=====
        //פעולה אשר מקבלת קוד ובודקת כמה פעמים מופיעים התווים
        //#!$
        //אם לפחות 2 יחזיר אמת אחרת מחזירה שקר
        public static bool Is2Digit(string code)
        // ABC
        {

```

```

    int Count = 0;
    // לולאה שעוברת על כל המחרוזת וסופרת את כמות התווים
    החייבים
    //#$!
    for (int i = 0; i < code.Length; i++)
    {
        if ((code[i] == '#') || (code[i] == '$') || (code[i] == '!'))
            Count++;
    }

    // כאן אנו בודקים אם הכמות היא לפחות 2 תווים ממה שהגדרנו
    אם כן תחזיר אמת אחרת שקר
    if (Count >= 2) return true;
    else
    {
        Console.WriteLine("ERROR you need at least 2
        digit form # $ !");
        return false;
    }
}

//=====
=====

    public static bool isLastCN(string code)
    {
        int code1 = code.Length; // מחזיר לי את אורך המחרוזת
        במספר
        if ((code[code1 - 2] == 'C') && (code[code1 - 1] ==
        'N')) // בדיקה אם שני התווים האחרונים הם סי ואן
            return true;
        else
        {
            Console.WriteLine("ERROR the 2 last digit in the
            code is not CN");
            return false;
        }
    }
}

```

```
//=====
=====
    // פעולה הבודקת אם הקוד בנוי מלפחות 8 תווים
    public static bool isLenAtLeast8Digit(string code)
    {
        int lenCode = code.Length; // מחזיר לי את אורך המחרוזת
        במספר
        if (lenCode >= 8)
            return true;
        else
        {
            Console.WriteLine("ERROR YOUR CODE MAST
            BE 8 DIGIT AT LEAST");
            return false;
        }
    }

//=====
=====

    static void Main(string[] args)
    {
        // הנה כאן הקוד לבדיקה
        string code = "cn#!kjhkjhCN";
        code = code.ToUpper(); // פונקציה אשר מגדילה את כל
        התווים במחרוזת לאותיות גדולות
        בדיקה של תנאי הראשון
        bool isTrue1 = IsFirestCN(code);
        Console.WriteLine("chek 1 = {0}", isTrue1);

        // בדיקה של התנאי השני
    }
}
```

```

bool isTrue2 = Is2Digit(code);
Console.WriteLine("chek 2 = {0}",isTrue2);

//בדיקה של התנאי אם שני התווים האחרונים הם סי ואן
bool isTrue3 = isLastCN(code);
Console.WriteLine("chek 3 = {0}", isTrue3);

//פעולה הבודקת את התנאי הרביעי אם אורך הקוד לפחות 8
תווים
bool isTrue4 = isLenAtLeast8Digit(code);
Console.WriteLine("chek 4 = {0}", isTrue4);

    }

}

}

```

שעורי בית : כתבו פעולה אשר מקבלת קוד , ומבצעת בדיקה של כל הפעולות ביחד , כך שאני אוכל לבדוק בתוכנית הראשית רק על ידי הקשה של שם פעולה אחד והקוד למשל
אם לפעולה נקרא ("ChekCode("cN345#\$ghCN) הוא יבדוק את הקוד ויחזיר לי תשובה אם הצליח או לא .

כתבו תוכנית אשר מקבלת מהמשתמש את הקוד שהוא רוצה לבחון, והמערכת בודקת את הקוד ומחזירה תשובה .

היום מפגש ביום ב בין השעות 14:30 - 23.03.2026

היום נתרגל המשך של פעולות , נדבר על הכרת טווח המשתנים
נתרגל פעולות שמקבלות גם פרמטרים יותר מורכבים כמו מערך .
בסוף השיעור תקבלו משימות לחופש לעד יום שני .

- א. נסיים את התרגיל שקבלתם בבית (פעולה אחת במקום כל הפקודות בתוכנית הראשית)
ב. לולאה שרצה עד אשר הקוד תקין .

ואז נעבור לשאלה חדשה להלן (שנפתור ביחד בשיעור) :
כתבו פעולה אשר מקבלת מערך של מספרים שלמים כלשהו , (גודל המערך יכול להשתנות) , הפעולה תסרוק את המערך ותחזיר את הערך המקסימלי שבו .

0	1	2	3	4	index
4	500	12	2	34	value

במקרה הזה הפעולה תחזיר : 500 שהוא הערך המקסימלי .

max	500
-----	-----

פתרון התרגיל של מציאת מקסימום בתוך מערך - פתרנו בכיתה

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Remoting.Channels;
using System.Security.Cryptography.X509Certificates;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp5
{
    internal class Program
    {
        //=====
        //פעולה המקבלת מערך ומחזירה את הערך המקסימלי שבו
        public static int maxNumArray(int[] array)
        {
            int maxNum = array[0]; //הוא המקסימלי
            for (int i = 1; i < array.Length; i++)
            {
                if (array[i] > maxNum) //בדיקה אם התא במערך הוא גדול מהמספר המקסימלי
                    maxNum = array[i]; //נעדכן שהמספר הגדול ביותר הוא התא הזה במערך
            }

            Console.WriteLine("the max number at this array is = {0}", maxNum);
            return maxNum;
        }

        static void Main(string[] args)
        {
            int[] arr1 = { 4, 5, 2, 78, 1023, 34, 2, 1, 788 }; //לד את אורך המערך
            //קריאה לפעולה אשר בודקת מי המספר הגדול
            //int maxNum = maxNumArray(arr1);

            Console.WriteLine(maxNumArray(arr1));
        }
    }
}
```

תרגיל כיתה :

עליכם לבנות פעולה המקבלת מערך של מספרים שלמים , ומחזירה ומדפיסה את האיבר הקטן ביותר בתוך המערך .

שיעורי בית לחופש

פלינדרום זו מחרוזת או מערך של תווים אשר קוראים אותו מהסוף להתחלה, או מההתחלה אל הסוף ויוצא לנו אותו המילה בדיוק

למשל אבא, אמא, לול, שמש,
 , "היפיפיה", 12321, !@##@!.

עליכם לכתוב פעולה אשר מקבלת מערך של מספרים שלמים בעל מספר איברים אי זוגי, ובודקת אם המערך הוא פלינדרום (תמונת ראי מאמצע המערך, כלומר אם סורקים אותו מהסוף להתחלה או מההתחלה לסוף הסריקה זהה).

אם הוא פלינדרום הפעולה תחזיר TRUE, אחרת FALSE

41	2	33	100	33	2	41
----	---	----	-----	----	---	----

יחזיר TRUE (כן פלינדרום)

=====

41	2	33	100	44	22	41
----	---	----	-----	----	----	----

יחזיר FALSE (לא פלינדרום)

שאלה מספר 2 - שאלת בונוס

חברת כח אדם ביקשה מעובדיה למסור לה קודים אישיים שמבקש כל עובד , קוד תקין הוא קוד בן 4 ספרות שאין בו רצף של 3 ספרות , ולא מתחיל ומסתיים באותה ספרה , כתבו קטע קוד שקולט סדרת קודים אישיים עד אשר נקלט קוד 0 , על התוכנית לחשב ולהדפיס מהו אחוז הקודים התקינים שהוקלדו .

יש לכתוב את התוכנית בעזרת שימוש בפעולות .

דוגמאות

הקוד	תקין / לא תקין	סיבה
12324565	לא תקין	שונה מ 4 ספרות
1234	לא תקין	מספרים בסדר עולה
1468	לא תקין	צריך להתחיל ולהסתיים באותה הספרה
1361	תקין	עונה על כל התנאים האחרים

להלן תוכנית של המשחק איקס עיגול שלמדנו בכיתה

שימו לב התוכנית מחולקת לפעולות
 אחרי יצירת מערך גלובלי (שמוכר בכל התוכנית)
 א. פעולה אשר מציגה את ערכי המערך ומציגה אותו בשלשות
 ב. פעולה המכניסה ערך (X או O) לתוך תא במערך
 ג. פעולה אשר מבצעת בדיקה האם יש ניצחון (ניצחון זה שיש שלשה
 אופקית שהם אותו התוו) .

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Remoting.Channels;
using System.Security.Cryptography.X509Certificates;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp5
{
    internal class Program
    {
        //פעולה המקבלת שני מספרים ומחזיר את הסכום ביניהם
        public static int sum2Num (int num1,int num2)
        {
            int tot = num1 + num2;
            return (tot);
        }
    }
}

//=====
```

```

===
public static void Printmenu()
{
    Console.WriteLine("=====");
    Console.WriteLine("apple1");
    Console.WriteLine("tpoz");
    Console.WriteLine("chasae");
    Console.WriteLine("avtich");
    Console.WriteLine("mango");
    Console.WriteLine("tot");
    Console.WriteLine("=====");

}

//=====
==
    פעולה שמדפיסה מספרים ממספר מסוים ועד מספר אחר/
    טענת הכניסה מספר התחל ומספר סופי/
    public static void count (int Start, int End)
    {

        for (int i = Start; i < End; i++)
        {
            Console.WriteLine(i);
        }

    }

    static void Main(string[] args)

    {
        count(5, 7); //פעולה המקבלת שני מספרים התחלה וסוף/
    }

```

ומדפיסה את המספרים ביניהם

פעולה המדפיסה את התפריט של הירקן//Printmenu();

```
    }
}
}
```

שיעור 15.4.2026 - מערך דו מימדי

היום נלמד על מערך דו מימדי

ונשלב את הידע בכתיבת פעולות אשר משמות אותנו לבניית של משחק מחשב מסוג איקס עיגול .

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Remoting.Channels;
using System.Security.Authentication.ExtendedProtection;
using System.Security.Cryptography.X509Certificates;
using System.Text;
using System.Threading.Tasks;
```

```
namespace ConsoleApp5
{
```

```
    internal class Program
    {
```

```
        static Char[] Numbers = new char[9];//הגדרה של משתנה
```

שכל התוכנית תכיר אותו

פעולה אשר מדפיסה את כל הערכים של//

כל התאים בתוך המערך

```
//=====
=====
```

פעולה אשר מדפיסה את ערכי המערך כלוח של 9 קוביות//

```
public static void PrintArr()
{
```

```
    for (int i = 0; i < Numbers.Length; i++) // שינוי ל-קטן ממש
    {
```

מדפיסים את התא הנוכחי בכל איטרציה //

```
        Console.WriteLine "[" + Numbers[i] + "];
```

בדיקה: אם הדפסנו 3 מספרים (אינדקסים 2, 5, 8), נרד //

שורה

```
        if ((i + 1) % 3 == 0)
```

```
        {
```

```
            Console.WriteLine();
```

```
        }
```

```
    }
```

```
}
```

```
//=====
=====
```

```
public static void InsertTav (int IndexTa,char Tav)
```

```
{
```

```
    Numbers[IndexTa]=Tav;
```

```
}
```

```
//=====
=====
```

```

//פעולה הבודקת האם יש ניצחון בתוך המערך
public static bool cheklfWin()
{
    bool Chk = false;

    if (Numbers[0]== Numbers[1]&&(Numbers[1]==
Numbers[2])
    {
        Console.WriteLine("the {0} is win",Numbers[0]);
        Chk= true;
    }

    return Chk;
}

static void Main(string[] args)
{
    PrintArr();//מצייר את המערך הריק
    InsertTav(4, 'X');//4 מכניס איקס למיקום האמצעי תא
    Console.WriteLine("++++++++++++++++");
    PrintArr();
    InsertTav(0, 'O');
    Console.WriteLine("++++++++++++++++");
    PrintArr();
    Console.WriteLine("++++++++++++++++");
}
}
}

```

בשיעור הבא נדבר על הנושא של טבלת ASCII

מיון בועות - מיון מערך

בשיעור זה נלמד ונכיר כיצד למיין את המערך , לדוגמה מקבלים מערך שבו יש ערכים שונים לא ממוינים , ואנו רוצים למיין את המערך .
 לשם כך נצטרך לבצע בדיקה מול שני אברי המערך ובמידה והם לא שומרים על הסדר נחליף ביניהם , וכך צריך לעבור על כל אברי המערך , שוב ושוב .

הנה הקוד שנעבור ונלמד אותו בשיעור

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace peola_array
{
    internal class Program
    {

//=====
=====

        //פעולה המקבלת מערך ומדפיסה את הערכים שלו זה ליד זה
        public static void PrintArray(int[] arrayToPrint)
        {
            Console.Write("[ ");
            for (int i = 0; i < arrayToPrint.Length; i++)
            {
                Console.Write(" {0} ", arrayToPrint[i]);
            }
            Console.WriteLine(" ]");
        }
    }
}
```

```
//=====
=====
```

```
public static int[] sortArray(int[] array)
{
    int lenAr = array.Length;
    int temp1=0;
    for (int i = 0; i < lenAr; i++)
    {
        for (int j = 0; j < lenAr-1; j++)
        {
            //בדיקה אם התא השמאלי גדול מהימני אז מחליפים
            במיקומים
            if (array[j] > array[j + 1])
            {
                temp1 = array[j];
                array[j] = array[j + 1];
                array[j + 1] = temp1;
            }
        }
    }
    return array; //מחזיר את המערך הממויין
}
```

```
//=====
=====
```

```
static void Main(string[] args)
{
    int[] ar1 = { 1, 7, 2, 9, 23, 56, 2, 5, 73, 12 };
    PrintArray(ar1); //הדפסת ערכי המערך החד ממדי
    Console.WriteLine("");
    PrintArray(sortArray(ar1)); //הדפסת ערכי המערך החד ממדי
}
```

ממדי

```

//sortArray(int[] array)

/*
הגדרנו מערך בעל 2 עמודות ומספר השורות 3
int[,] grid = {
    {10,20,34,4 },
    {30,40,1,6},
    {50,60 ,8,7}
};
//לולאה חיצונית הסופרת את מספרי השורות
for (int i = 0; i < grid.GetLength(0); i++)
{
    //לולאה פנימית הסופרת את מספר העמודה לאותה שורה
    for (int j = 0; j < grid.GetLength(1); j++)
    {
        Console.Write(grid[i,j]+" ");
    }
    Console.WriteLine();
}
*/
}
}
}

```

תרגיל כיתה מורכב

עליכם לכתוב תוכנית מחשב אשר קולטת נתונים עבור ספורטאים
בראשית התוכנית קולטת עבור כל ספורטאי את סוג הענף שבו הוא
משתתף

עבור ענף ריצה יש להקיש על התוו R
עבור ענף קפיצה לגובה יש להקיש על התוו J
לאחר מכן יש להקיש את התוצאה שהשיג הספוראי באותו הענף .

התוכנית תקלוט נתונים של שחקנים עד אשר המשתמש יקיש תוו שונה מ
 R ושונה מ J .

בסיום התוכנית התוכנית תדפיס
מהי התוצאה הגבוהה ביותר של ענף הריצה R
ומהי התוצאה הגבוהה ביותר של ענף הקפיצה J .

הנה פתרון התרגיל שלי

העתיקו את התוכנית שלי למחשב ועקבו אחריה - נחזור על כך בכיתה

```
using System;
namespace sportProgram1
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int maxRun = 0;
            int MaxJump=0;
            //הדפסת הנחיות למשתמש
            Console.WriteLine("sport program ");
            Console.WriteLine("you will insert the anaf of
sportMan (R or J) and his score");

            //קליטת הענף של הספוראי
            Console.WriteLine("insert the ANAF");
            char Tav =
char.ToUpper(Console.ReadKey().KeyChar);

            if ((Tav == 'R') || (Tav == 'J'))
            {

                //קליטת התוצאה של הספורטאי
                Console.WriteLine("insert the Score");

                //קליטת נתון לריצה
                if (Tav == 'R')
                {
                    maxRun = int.Parse(Console.ReadLine());
                }
            }
        }
    }
}
```

```

        קליטה נתון לקפיצה לגובה
        if (Tav == 'J')
        {

            MaxJump = int.Parse(Console.ReadLine());
        }

        do
        {

            קליטת הענף של הספראי
            Console.WriteLine("insert the ather ANAF");
            Tav =
char.ToUpper(Console.ReadKey().KeyChar);

            הפיכת התוו שקלטנו
            // Tav = char.ToUpper(Tav);
            תמיד לאות גדולה
            //
            if (Tav != 'R' && Tav != 'J')
                break;

            Console.WriteLine("insert the ather Score");
            int Score = int.Parse(Console.ReadLine());
            קליטת נתון לריצה
            if (Tav == 'R')
            {
                if (Score > maxRun)
                    maxRun=Score;
            }

            קליטה נתון לקפיצה לגובה
            if (Tav == 'J')
            {
                if (Score > MaxJump)
                    MaxJump = Score;
            }
        }
    
```

```
    }  
    while (Tav == 'R' || Tav == 'J');  
  
    }  
    else  
    {  
        Console.WriteLine("end no sport man ");  
    }  
  
    Console.WriteLine("the maxRun is {0} ", maxRun);  
    Console.WriteLine("the maxJumpn is {0} ", MaxJump);  
  
    }  
    }  
}
```

שיעור 17.5.2026

תרגיל מספר 1 - כתוב תוכנית מחשב אשר מבצעת פעולות חשבון (מחשבון)

א. התוכנית תציג תפריט של האפשרויות המותרות

1. חיבור בין שני מספרים
2. חיסור בין שני מספרים
3. כפל בין שני מספרים
4. חילוק בין שני מספרים
5. שורש ריבועי של המספר
6. העלאה בחזקה של המספר

כל תת סעיף יתבצע על ידי פעולה שמקבלת 2 מספרים ומחזירה את התוצאה .

לדוגמה המשתמש בוחר באפשרות 1, והקיש מספר ראשון ושני והמחשב יציג לו את תוצאת החיבור ביניהם .

תרגיל מספר 2

כתבו תוכנית מחשב (שעשינו בעבר דומה לזה) אשר המחשב מגריל מספר בין 1 לבין 100 , והמשתמש צריך לנחש את המספר המוקש

במידה והמשתמש הקיש את התוצאה שהמחשב ניחש - התוכנית תדפיס כל הכבוד ניצחת ב X ניסיונות .

במידה והמשתמש הקיש מספר גדול יותר מהמחשב - התוכנית תדפיס המספר שהקשת גדול מהמספר של המחשב ותמשיך .

במידה והמשתמש הקיש מספר קטן יותר מהמחשב - התוכנית תדפיס
המספר שהקשת קטן מהמספר של המחשב
ותמשיך

מתקדם :

א. רשום את מספר האפשרויות הניתן , ובכל פעם הורד את האפשרות ב 1
והצג למשתמש את מספר הניחושים שנותרו לו . ברגע שסיים את כולם בלי
לנחש את התוצאה , המשחק יסתיים ויקבל הודעה "הפסדת" אחרת יקבל
הודעה ניצחת ב (מספר הניחושים שניחש) .

ב. הוסף פעולה אשר מקבלת שני משתנים מספר נמוך ועד איזה מספר ,
ומבצעת הגרלה של המספר בן המספרים האלו .