

פעולה (שגרה/פרוצדורה) בשפת C עם פרמטרים

ומחזירה ערך

מה זו פעולה שמקבלת פרמטרים ומחזירה ערך בשפת C?#

פעולה שמקבלת פרמטרים ומחזירה ערך מבצעת משימה על סמך הנתונים שהועברו אליה, ולאחר מכן מחזירה תוצאה למי שקרא לה. תוצאה זו יכולה להיות מסוג נתון אחד כמו `int`, `string`, או אפילו אובייקט מותאם אישית.

מטרת פעולות שמחזירות ערכים

1. לספק תוצאות לחישוב – פעולות אלו מחזירות את התוצאה של המשימה שביצעו.
2. להגביר את השימושיות של הקוד – אפשר לקרוא לפעולה ולשמור את התוצאה במשתנה, או להשתמש בה מיד.
3. שימוש חוזר בקוד – במקום לכתוב את אותה פעולה שוב ושוב, משתמשים בפעולה אחת שמתאימה לצרכים משתנים.

דוגמה לפעולה שמקבלת פרמטרים ומחזירה ערך

דוגמה 1: פעולה שמקבלת שני מספרים, שמחשבת סכום ומחזירה אותו

```
using System;

class Program
{
    // פעולה שמחזירה את סכום שני מספרים
    static int CalculateSum(int num1, int num2)
    {
        return num1 + num2;
    }

    static void Main(string[] args)
    {
        int sum = CalculateSum(10, 20); // קריאה לפעולה ושמירת התוצאה במשתנה
        Console.WriteLine($"הסכום הוא: {sum}");
    }
}
```

הסבר על הקוד

1. `CalculateSum` – שם הפעולה.
2. הפעולה מקבלת שני פרמטרים (`num1`, `num2`) מסוג `int`.
3. הפעולה מחזירה תוצאה מסוג `int` באמצעות הפקודה `.return`.
4. התוצאה נשמרת במשתנה בשם `sum` ב-`Main`.

דוגמה 2: פעולה שבודקת אם מספר זוגי ומחזירה תשובה

```

class Program
{
    // פעולה שמחזירה אם מספר זוגי או לא
    static bool IsEven(int number)
    {
        return number % 2 == 0;
    }

    static void Main(string[] args)
    {
        int num = 7;
        bool result = IsEven(num); // קריאה לפעולה ושמירת התוצאה במשתנה

        if (result)
            Console.WriteLine($"{num} הוא מספר זוגי");
        else
            Console.WriteLine($"{num} הוא מספר אי-זוגי");
    }
}

```



תוצאה

.הוא מספר אי-זוגי 7

דוגמה 3 : פעולה שמחשבת ממוצע של מספרים

```
using System;

class Program
{
    // פעולה שמחשבת ממוצע של שלושה מספרים
    static double CalculateAverage(int num1, int num2, int num3)
    {
        double sum = num1 + num2 + num3;
        return sum / 3; // חישוב הממוצע
    }

    static void Main(string[] args)
    {
        // קריאה לפעולה
        double average = CalculateAverage(10, 20, 30);

        // הדפסת התוצאה
        Console.WriteLine($"הממוצע של המספרים הוא: {average}");
    }
}
```



הסבר על הקוד

הפעולה: CalculateAverage

1. שם הפעולה: CalculateAverage – מתאר את מטרת הפעולה: לחשב ממוצע.
2. הפרמטרים:
 - `int num1, int num2, int num3` – שלושה מספרים מסוג `int` שהמשתמש מספק לפעולה.
3. החישוב:
 - מחברים את המספרים עם `num1 + num2 + num3` ושומרים במשתנה `.sum`.
 - מחשבים את הממוצע על ידי חלוקת הסכום ב-3.
4. הערך המוחזר:
 - הפעולה מחזירה את התוצאה (הממוצע) מסוג `double`.

הקריאה לפעולה ב-Main

- `double average = CalculateAverage(10, 20, 30);`
 - פעולה זו מעבירה את המספרים 10, 20 ו-30 כפרמטרים לפעולה.
 - התוצאה מחושבת ונשמרת במשתנה `average`.

תוצאה

הממוצע של המספרים הוא: 20

שימושים רלוונטיים

1. גמישות:
 - הפעולה מאפשרת לחשב ממוצע לכל שלושה מספרים שמועברים כפרמטרים.
2. שימוש חוזר:
 - במקום לכתוב את נוסחת הממוצע שוב ושוב, הפעולה מרכזת את החישוב במקום אחד.

שדרוגים אפשריים

1. ממוצע של מספרים רבים:

- ניתן לשנות את הפעולה כך שתקבל מערך של מספרים ותחשב ממוצע עבור כל הערכים שבו.

2. תמיכה במספרים עשרוניים:

- לשנות את הפרמטרים מסוג `int` ל-`double`, כך שהפעולה תוכל לטפל גם במספרים עשרוניים.

דוגמה 4 : פעולה שמקבלת שלושה ערכים ומחזירה שני ערכים

פעולה שמקבלת שלושה מספרים ומחשבת את הסכום והממוצע של שלושתם

```
using System;

class Program
{
    // פעולה שמחזירה סכום וממוצע של שלושה מספרים
    static (int sum, double average) CalculateSumAndAverage(int num1, int num2, int num3)
    {
        int sum = num1 + num2 + num3; // חישוב הסכום
        double average = sum / 3.0; // חישוב הממוצע
        return (sum, average); // החזרת שני ערכים
    }

    static void Main(string[] args)
    {
        // קריאה לפעולה
        var result = CalculateSumAndAverage(10, 20, 30);

        // הצגת התוצאות
        Console.WriteLine($"הסכום הוא: {result.sum}");
        Console.WriteLine($"הממוצע הוא: {result.average}");
    }
}
```



הסבר על הקוד

הפעולה: `CalculateSumAndAverage`

1. שם הפעולה: `CalculateSumAndAverage` – מתאר את תפקידה של הפעולה.
2. הפרמטרים:
 - `int num1, int num2, int num3` – שלושה מספרים שהפעולה מקבלת כקלט.
3. החישוב:
 - `sum` – מחושב באמצעות חיבור שלושת המספרים.
 - `average` – מחושב על ידי חלוקת הסכום ב-3.0 כדי להבטיח תוצאה עשרונית (`double`).
4. הערך המוחזר:
 - הפעולה מחזירה שני ערכים באמצעות **טאפל** (קבוצה): `sum` ו-`average`.

הקריאה לפעולה ב-`Main`

```
var result = CalculateSumAndAverage(10, 20, 30);
```

- הפעולה מחזירה קבוצה (`tuple`) שמכילה את הסכום והממוצע.
- התוצאה נשמרת במשתנה `result`.

שימוש בערכים המוחזרים

- כדי לגשת לערכים:
 - `result.sum` – הסכום.
 - `result.average` – הממוצע.

תוצאה

הסכום הוא: 60
הממוצע הוא: 20

שימושים רלוונטיים

1. יעילות בקוד:

○ החזרת כמה ערכים באותו זמן חוסכת יצירת פעולות נפרדות.

2. ארגון נתונים:

○ שימוש בטאפל עוזר לארגן ולנהל את הערכים המוחזרים בצורה פשוטה

וברורה.

שדרוגים אפשריים

1. שימוש במערכים:

○ ניתן להחזיר יותר ערכים על ידי שימוש במבנים כמו רשימות או מערכים.

2. שימוש באובייקטים:

○ אפשר להחזיר אובייקט שמכיל את כל המידע, אם נדרש יותר גמישות ותחזוקה

ארוכת טווח.

תרגילים לכתיבת פעולות שמקבלות פרמטרים ומחזירות ערך/כים

1. בדיקת מספר זוגי

- כתוב פעולה שמקבלת מספר שלם כפרמטר ובודקת אם הוא זוגי.
- הפעולה תחזיר **true** אם המספר זוגי, ו-**false** אם לא.

2. חישוב שטח והיקף מלבן

- כתוב פעולה שמקבלת שני פרמטרים (אורך ורוחב של מלבן) ומחזירה שני ערכים:
 - שטח המלבן.
 - היקף המלבן.

3. מספר מקסימלי ומינימלי

- כתוב פעולה שמקבלת שלושה מספרים שלמים ומחזירה:
 - את המספר הגדול ביותר.
 - את המספר הקטן ביותר.

4. חישוב תוצאה במבחן

- כתוב פעולה שמקבלת שלושה ציונים של מבחנים ומחזירה:
 - את ממוצע הציונים.
 - אינדיקציה (**true/false**) אם התלמיד עבר (ממוצע מעל 60).

5. חישוב שורש ריבועי ● כתוב פעולה שמקבלת מספר חיובי ומחזירה את השורש הריבועי שלו.
6.