

מערכים

מערכים – מעבר על מערך

// מערכים - מעבר על מערך
// 20. כתוב פעולה המקבלת מערך מסוג מספר שלם ומחזירה את סכום האיברים במערך

```
public static int SumOfArr(int[] arr)
{
    int sum = 0;
    for (int i = 0; i < arr.Length; i++)
    {
        sum = sum + arr[i];
    }
    return sum;
}
```

// 21. כתוב פעולה המקבלת מערך מסוג מספר שלם ומחזירה את ממוצע האיברים במערך

```
public static int AvgOfArr(int[] arr)
{
    int sum = 0;
    int count = 0;
    for (int i = 0; i < arr.Length; i++)
    {
        count++;
        sum = sum + arr[i];
    }
    return sum / count;
}
```

// 22. ומחזירה האם NUM כתוב פעולה המקבלת מערך מסוג מספר שלם ועוד מספר שלם מופיע הNUM המספר במערך

```
public static bool NumExist(int[] arr, int num)
{
    for (int i = 0; i < arr.Length; i++)
    {
        if (arr[i] == num)
        {
            return true;
        }
    }
    return false;
}
```

// 23. כתוב פעולה המקבלת מערך מסוג מספר שלם, ומחזירה כמה מספרים זוגיים יש במערך.

```
public static int evenNumbers(int[] arr)
{
    int count_of_even = 0;
    for (int i = 0; i < arr.Length; i++)
    {
        if (arr[i] % 2 == 0)
        {
            count_of_even++;
        }
    }
    return count_of_even;
}
```

// 24. כתוב פעולה המקבלת מערך מסוג מספר שלם, ומחזירה האם יש לפחות 5 מספרים זוגיים

```
public static bool fiveEvenNumbers(int[] arr)
{
    int count_of_even = 0;
    for (int i = 0; i < arr.Length; i++)
    {
        if (arr[i] % 2 == 0)
        {
            count_of_even++;

            if (count_of_even >= 5)
            {
                return true;
            }
        }
    }
    return false;
}
```

// 25. כתוב פעולה המקבלת מערך מסוג מספר שלם, ומחזירה כמה מספרים גדולים מהמוצע של המערך.

```
public static int biggerThenAverage(int[] arr)
{
    int sum = 0;
    int avg = 0;
    int count = 0;
    for (int i = 0; i < arr.Length; i++)
    {
        sum += arr[i];
    }
    avg = sum / arr.Length;

    for (int i = 0; i < arr.Length; i++)
    {
        if (arr[i] > avg)
        {
            count++;
        }
    }
    return count;
}
```

// 26. ומחזירה האם NUM כתוב פעולה המקבלת מערך מסוג מספר שלם ועוד מספר שלם. כל המערך הוא המספר

// NUM

```
public static bool fullOfNums(int[] arr, int num)
{
    for (int i = 0; i < arr.Length; i++)
    {
        if (arr[i] != num)
        {
            return false;
        }
    }
    return true;
}
```

// 27. כתוב פעולה המקבלת מערך מסוג מספר שלם, ומחזירה את המספר הכי גדול במערך

```
public static int theBigestNum(int[] arr)
{
    int max = arr[0];
    for (int i = 0; i < arr.Length; i++)
    {
        if (arr[i] > max)
        {
            max = arr[i];
        }
    }
    return max;
}
```

// 28. כתוב פעולה המקבלת מערך מסוג מספר שלם, ומחזירה את המיקום של המספר הכי גדול במערך (יש רק מספר אחד שהוא הכי גדול)

```
public static int theIndexOfTheMax(int[] arr)
{
    int index = 0;
    int max = arr[0];
    for (int i = 0; i < arr.Length; i++)
    {
        if (arr[i] > max)
        {
            max = arr[i];
            index = i;
        }
    }
    return index;
}
```

// 29. כתוב פעולה המקבלת מערך מסוג מספר שלם, ומחזירה את המספר הכי קטן במערך

```
public static int theSmallestNumber(int[] arr)
{
    int min = arr[0];
    for (int i = 0; i < arr.Length; i++)
    {
        if (arr[i] < min)
        {
            min = arr[i];
        }
    }
    return min;
}
```

// 30. כתוב פעולה המקבלת מערך מסוג מספר עשרוני ומחזירה את המקום של המספר הכי קרוב ל0

```
public static double theMostClosedTo0(double[] arr)
{
    double min = arr[0];
    for (int i = 0; i < arr.Length; i++)
    {
        if (arr[i] < min)
        {
            min = arr[i];
        }
    }
}
```

```

        return min;
    }

    // 31. כתוב פעולה המקבלת 2 מערכים מסוג מספר שלם, ומחזירה האם הם בדיוק
    // אותו הדבר, באותו גודל ועם אותו
    // תוכן לפי אותו הסדר
    public static bool theBiggestArr(int[] arr1, int[] arr2)
    {
        if (arr1.Length == arr2.Length)
        {
            for (int i = 0; i < arr1.Length; i++)
            {
                if (arr1[i] == arr2[i])
                {
                    for (int j = 0; j < arr1.Length; j++)
                    {
                        if (arr1[j] + 1 == arr2[j] + 1)
                        {
                            return true;
                        }
                    }
                }
            }
        }
        return false;
    }
}

```

```

    // 32. כתוב פעולה המקבלת 2 מערכים מסוג מספר שלם, ומחזירה האם יש מספר
    // שמופיע ב-2 המערכים
    public static bool sameNumberInTheTwoArrays(int[] arr1, int[] arr2)
    {
        for (int i = 0; i < arr1.Length; i++)
        {
            if (arr1[i] != arr2[i])
            {
                return false;
            }
        }
        return true;
    }
}

```

מערכים – מעבר מהסוף

```

    // 40. כתוב פעולה המקבלת מערך מסוג מספר שלם ומדפיסה את המספרים מסוף המערך
    // עד ההתחלה
    public static void lastToFirst(int[] arr)
    {
        for (int i = arr.Length-1; i >= 0; i--)
        {
            Console.WriteLine(arr[i]);
        }
    }
}

```

```

    // 41. ומחזירה את NUM, כתוב פעולה המקבלת מערך מסוג מספר שלם, ומספר שלם
    // ש NUM המיקום האחרון ש
    // לא מופיע במערך, יש להחזיר 1- NUM מופיע במערך, אם
    public static int theLastIndexNumber(int[] arr, int num)
    {

```

```

int lastIndex = -1;
for (int i = 0; i < arr.Length; i++)
{
    if (arr[i] == num)
    {
        lastIndex = i;
    }
}
return lastIndex;
}

// 42. הפעולה מפנה NUM, כתוב פעולה המקבלת מערך מסוג מספר שלם, ומספר שלם.
// בתחילת NUM מקום ל
// נכנס למקום 0, מה שהיה במקום NUM המערך, ודוחפת את כל האיברים קדימה. ז"א
// 0 עבשיו יהיה במקום 1,
// 'שהיה במקום 1 עבשיו יהיה במקום 2, וכו'
public static int[] insertNumberToStart(int[] arr, int num)
{
    int[] newArr = new int[arr.Length + 1];
    newArr[0] = num;
    for (int i = 1; i < newArr.Length; i++)
    {
        newArr[i] = arr[i - 1];
    }
    return newArr;
}

```

מערכים - סכומים חלקיים/מעבר חלקי

```

// 50. כתוב פעולה המקבלת מערך מסוג מספר שלם ובודקת האם סכום החצי הראשון
// של המערך שווה לחצי השני
// (של המערך) מספר האיברים במערך הוא זוגי
public static bool sumOfHalf(int[] arr)
{
    int sum1 = 0;
    int sum2 = 0;
    for (int i = 0; i < arr.Length / 2; i++)
    {
        sum1 = sum1 + arr[i];
    }

    for (int i = 0; i > arr.Length / 2; i++)
    {
        sum2 = sum2 + arr[i];
    }

    if (sum1 == sum2)
    {
        return true;
    }
    return false;
}

```

```

// 51. ומחזירה NUM כתוב פעולה המקבלת מערך מסוג מספר שלם ועוד מספר שלם.
// האיברים NUM האם סכום
// הראשון גדול מסכום שאר האיברים.
public static bool sumOfNums(int[] arr, int num)
{
    int sum1 = 0;
    int sum2 = 0;
    for (int i = 0; i < num; i++)
    {

```

```

        sum1 = sum1 + arr[i];
    }

    for (int i = 0; i < num; i++)
    {
        sum2 = sum2 + arr[i];
    }

    if (sum1 > sum2)
    {
        return true;
    }
    return false;
}

```

// 52. כתוב פעולה המקבלת מערך מסוג מספר שלם, ומחזירה את המיקום שבו סכום האיברים לפני שווה לסכום האיברים אחרי (לא כולל אותו המקום) אם אין מקום כזה, יש להחזיר -1. יש (לפחות 3 איברים במערך)

```

public static int theLocationOfTheIndex(int[] arr)
{
    int leftHalf = arr[0];
    int rightHalf = 0;
    for (int i = 1; i < arr.Length; i++)
    {
        for (int j = i + 1; j < arr.Length; j++)
        {
            rightHalf = rightHalf + arr[j];
        }
        Console.WriteLine("lefthalf -----" + leftHalf);
        Console.WriteLine("righthalf -----" + rightHalf);
        if (leftHalf == rightHalf)
        {
            return i;
        }
        leftHalf = leftHalf + arr[i];
        rightHalf = 0;
    }
    return -1;
}

```

// 53. כתוב פעולה המקבלת מערך מסוג מספר שלם, והופכת את הסדר במערך.
 // דוגמה: המערך לפני: 5 4 3 2 1
 // אחרי: 1 2 3 4 5

```

public static int[] reverseArr(int[] arr)
{
    int temp;
    for (int i = 0; i < arr.Length / 2; i++)
    {
        temp = arr[arr.Length - 1 - i];
        arr[arr.Length - 1 - i] = arr[i];
        arr[i] = temp;
    }
    return arr;
}

```

מערכים – מיונים

// 60. כתוב פעולה המקבלת מערך מסוג מספר שלם ומחזירה האם המערך ממוין בסדר עולה

```

public static bool IsAscending(int[] arr)

```

```

{
    for (int i = 1; i < arr.Length; i++)
    {
        if (arr[i] < arr[i - 1])
        {
            return false;
        }
    }
    return true;
}

```

// 61. כתוב פעולה המקבלת מערך מסוג מספר שלם של ציונים, ומחזירה האם יש שיפור במשך 3 ציונים. ז"א האם

(יש 3 מספרים רצופים בסדר עולה במערך) יש לפחות 3 מספרים במערך

```

public static bool IsThreeAscending(int[] arr)
{
    for (int i = 2; i < arr.Length; i++)
    {
        if (arr[i - 2] < arr[i - 1] && arr[i - 1] < arr[i])
        {
            return true;
        }
    }
    return false;
}

```

// 62. יש num כתוב פעולה המקבלת מערך ממויין מסוג מספר שלם ומספר שלם num, להכניס את למקומו במערך num להכניס את

בי במקום האחרון במערך יש אפס num יש מקום בשביל המספר

```

public static int[] AscendingWithNumber(int[] arr, int num)
{
    int index = 0;
    for (int i = 1; i < arr.Length; i++)
    {
        if (num > arr[i - 1] && num < arr[i])
        {
            index = i;
            break;
        }
    }

    for (int i = arr.Length - 1; i > index; i--)
    {
        arr[i] = arr[i - 1];
    }
    arr[index] = num;
    return arr;
}

```

// לעבור קצת עם אסף מבאן אולי

// 63. כתוב פעולה המקבלת מערך מסוג מספר שלם, וממיינת אותו

```

public static void SortingArray(int[] arr)
{
    int min = 0;
    int index = 0;
    for (int i = 0; i < arr.Length; i++)
    {
        min = arr[i];
        index = i;
        for (int j = i + 1; j < arr.Length; j++)
        {

```

```

        if (min > arr[j])
        {
            min = arr[j];
            index = j;
        }
    }
    int temp = arr[i];
    arr[i] = min;
    arr[index] = temp;
}
}

```

// 64. כתוב פעולה המקבלת 2 מערכים ממויינים מסוג מספר שלם, ומחזירה מערך חדש ממויין עם המספרים מ2 המערכים.

```

public static int[] MergeAndSort(int[] arr1, int[] arr2)
{
    int[] arr = new int[arr1.Length + arr2.Length];
    for (int i = 0; i < arr1.Length; i++)
    {
        arr[i] = arr1[i];
    }
    for (int i = 0; i < arr2.Length; i++)
    {
        arr[arr1.Length + i] = arr2[i];
    }
    SortingArray(arr);
    return arr;
}

```

מערכים – גלישה ממערך

// מערכים – גלישה ממערך
 // 70. כתוב פעולה המקבלת מערך מסוג מספר שלם ומחזירה האם הרצף 1,2,3 מופיע במערך

```

public static bool OneTwoThreeInTheArray(int[] arr)
{
    for (int i = 0; i < arr.Length - 2; i++)
    {
        if (arr[i] == 1 && arr[i + 1] == 2 && arr[i + 2] == 3)
        {
            return true;
        }
    }
    return false;
}

```

// 72. כתוב פעולה המקבלת מערך מסוג מספר שלם ומחזירה האם המערך הוא סדרת פיבונצ'י (כל איבר הוא סכום 2 האיברים הקודמים), מספר האיברים במערך הוא לפחות 3

```

public static bool Fibonacci(int[] arr)
{
    if (arr.Length < 3)
    {
        return false;
    }

    for (int i = 2; i < arr.Length; i++)
    {
        if (arr[i] != arr[i - 1] + arr[i - 2])
        {

```



```

        return false;
    }
}
return true;
}

// 73. כתוב פעולה המקבלת מערך מסוג מספר שלם, ומחפשת את רצף המספרים הכי גדול. יש להדפיס מה המספר שיש לו את הרצף הכי גדול, וכמה איברים יש ברצף
// א.אפשר להניח שיש רק רצף אחד שהוא הכי גדול
public static void TheBiggestSequence(int[] arr)
{
    int count = 1;
    int maxCount = 1;
    int maxNum = arr[0];
    int startSeq = 0;
    for (int i = 1; i < arr.Length; i++)
    {
        if (arr[i] == arr[i - 1] + 1)
        {
            count++;
        }
        else
        {
            if (count > maxCount)
            {
                maxCount = count;
                maxNum = arr[i - 1];
                startSeq = i - 1;
            }
            count = 1;
        }
    }

    if (count > maxCount)
    {
        maxCount = count;
        maxNum = arr[arr.Length - 1];
    }
    Console.WriteLine("The biggest sequence is: " + maxCount + " and the number is: " + maxNum);
    for (int i = startSeq - maxCount + 1; i < startSeq + 1; i++)
    {
        Console.WriteLine(arr[i]);
    }
}

```

מערכים – בניית מערכים

// 80. כתוב פעולה המקבלת מערך של מספר שלם, ומחזירה מערך חדש שהוא שכפול של המערך שקיבלנו.

```

public static int[] DoubleArray(int[] arr)
{
    int[] NewArr = new int[arr.Length];
    for (int i = 0; i < NewArr.Length; i++)
    {
        NewArr[i] = arr[i];
    }
    return NewArr;
}

```

// 81. במערך יש NUM ומחזירה מערך בגודל NUM, כתוב פעולה המקבלת מספר שלם.
להכניס את סדרת פיבונאצ'י
//)גודל או שווה 2 NUM המספר(

```
public static int[] NumFibonachi(int num)
{
    int[] NewArr = new int[num];
    NewArr[0] = 0;
    NewArr[1] = 1;
    for (int i = 2; i < num; i++)
    {
        NewArr[i] = NewArr[i - 1] + NewArr[i - 2];
    }
    return NewArr;
}
```

// 82. כתוב פעולה המקבלת מערך מסוג מספר שלם, ומחזירה מערך בגודל 3 במקום הראשון יש להכניס כמה מספרים זוגיים יש במערך, במקום השני יש להכניס כמה מספר אי זוגיים יש בערך ובמקום השלישי יש להכניס כמה אפסים יש במערך.

```
public static int[] ArrayLengh3(int[] arr)
{
    int[] NewArr = new int[3];

    int CountOfEven = 0;
    int CountOfOdd = 0;
    int CountOfZeroes = 0;

    for (int i = 0; i < arr.Length; i++)
    {
        if (arr[i] % 2 == 0)
        {
            CountOfEven++;
        }

        if (arr[i] % 2 == 1)
        {
            CountOfOdd++;
        }

        if (arr[i] == 0)
        {
            CountOfZeroes++;
        }
    }

    NewArr[0] = CountOfEven;
    NewArr[1] = CountOfOdd;
    NewArr[2] = CountOfZeroes;
    return NewArr;
}
```

// 83. כתוב פעולה המקבלת מערך מסוג מספר שלם, ומחזירה מערך חדש שבו מופיעים רק המספרים הזוגיים

```
public static int[] EvenArray(int[] arr)
{
    int CountOfEven = 0;
    for (int i = 0; i < arr.Length; i++)
    {
        if (arr[i] % 2 == 0)
        {
            CountOfEven++;
        }
    }
}
```

```

    }
}

int index = 0;
int[] NewArr = new int[CountOfEven];
for (int i = 0; i < arr.Length; i++)
{
    if (arr[i] % 2 == 0)
    {
        NewArr[index] = arr[i];
        index++;
    }
}
return NewArr;
}

```

מערכים – מערך מונים/צוברים

// מערכים - מערך מונים/צוברים
 // 90. כתוב פעולה המקבלת מערך מסוג מספר שלם שבו ומחזירה האם כל המספרים מ 0 עד 10 מופיעים במערך

```

public static bool TenToZero(int[] arr)
{
    int[] CounterArr = new int[11];
    for (int i = 0; i < arr.Length; i++)
    {
        if (arr[i] >= 0 && arr[i] <= 10)
        {
            CounterArr[arr[i]]++;
        }
    }

    for (int i = 0; i < CounterArr.Length; i++)
    {
        if (CounterArr[i] == 0)
        {
            return false;
        }
    }
    return true;
}

```

// 91. כתוב פעולה המקבלת מערך מסוג מספר שלם עם מספרים מ 0 עד 10, ומחזירה את המספר שמופיעה הכי הרבה פעמים במערך.

```

public static int MostCommonNumberBetweenZeroToTen(int[] arr)
{
    int[] CounterArr = new int[11];
    for (int i = 0; i < arr.Length; i++)
    {
        if (arr[i] >= 0 && arr[i] <= 10)
        {
            CounterArr[arr[i]]++;
        }
    }

    int max = CounterArr[0];
    int index = 0;
    for (int i = 1; i < CounterArr.Length; i++)
    {
        if (CounterArr[i] > max)
        {

```

```

        max = CounterArr[i];
        index = i;
    }
}
return index;
}

```

// 92. כתוב פעולה המקבלת מערך מסוג מספרים שלמים בגודל 100, ומחזירה לאיזה עשירון יש את הסכום הגדול ביותר. העשירון הראשון אלו התאים מ0 עד 9 (כולל), העשירון השני אלו התאים מ10 עד 19 (כולל), וכו'.

```

public static int TheRichestDecile(int[] arr)
{
    int[] CounterArr = new int[10];
    for (int i = 0; i < arr.Length; i++)
    {
        CounterArr[arr[i] / 10] += arr[i];
    }

    int max = CounterArr[0];
    int index = 0;
    for (int i = 1; i < CounterArr.Length; i++)
    {
        if (CounterArr[i] > max)
        {
            max = CounterArr[i];
            index = i;
        }
    }
    return index + 1;
}

```

// 93. כתוב פעולה המקבלת מערך של 360 מדידות של גשם, ויש להחזיר באיזה חודש ירד הכי הרבה גשם. כל חודש עם 30 ימים.

```

public static int TheMostRainyMonth(int[] arr)
{
    int[] CounterArr = new int[12];
    for (int i = 0; i < arr.Length; i++)
    {
        CounterArr[i / 30] += arr[i];
    }

    int max = CounterArr[0];
    int index = 0;
    for (int i = 1; i < CounterArr.Length; i++)
    {
        if (CounterArr[i] > max)
        {
            max = CounterArr[i];
            index = i;
        }
    }
    return index + 1;
}

```