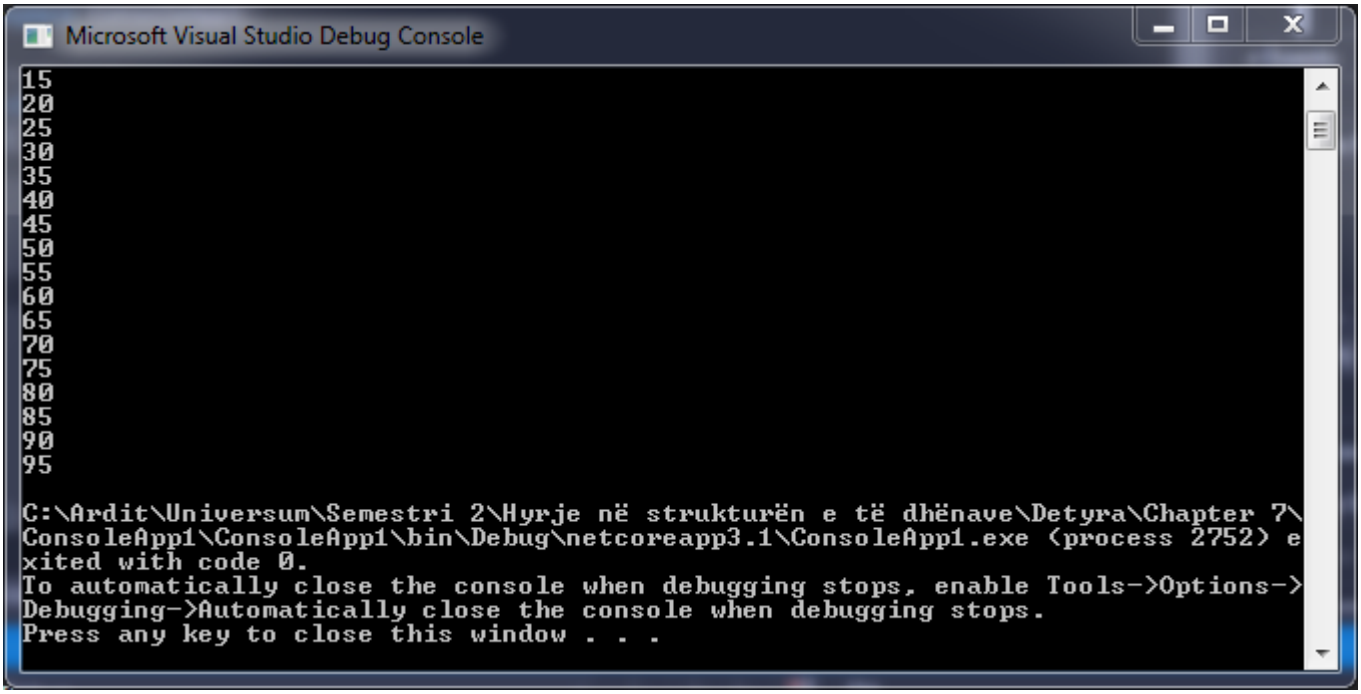
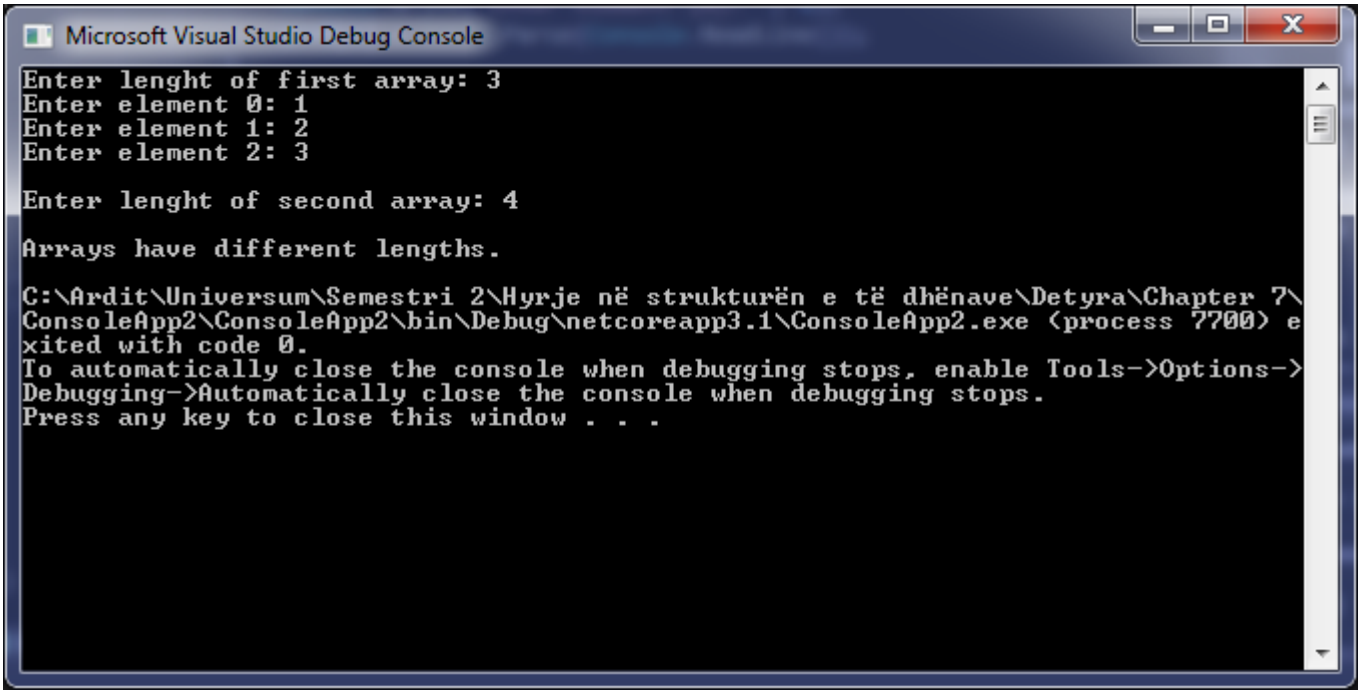
1. Write a program, which creates an array of **20 elements of type integer** and initializes each of the elements with a value equals to the index of the element multiplied by 5. Print the elements to the console.

```
using System;

namespace ConsoleApp1
{
    class Program
    {
        static void Main(string[] args)
        {
            int[] arr = new int[20];

            for (int i = 0; i < arr.Length; i++)
            {
                arr[i] = i * 5;
                Console.WriteLine(arr[i]);
            }
        }
    }
}
```



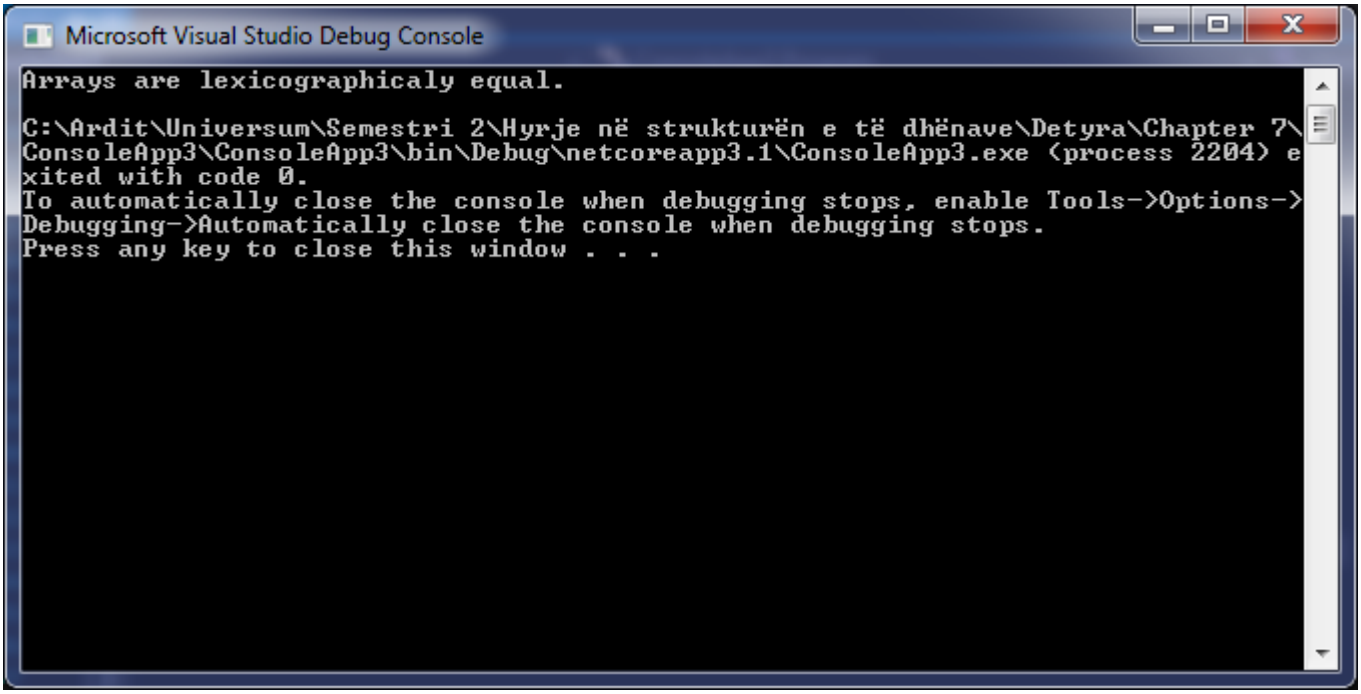
2. Write a program, which **reads two arrays** from the console and **checks whether they are equal** (two arrays are equal when they are of equal length and all of their elements, which have the same index, are equal).

```
using System;

namespace ConsoleApp2
{
    class Program
    {

        static void Main(string[] args)
        {
            bool arraysEqual = true;

            Console.Write("Enter lenght of first array: ");
            int length = Int32.Parse(Console.ReadLine());

            int[] arrA = new int[length];

            for (int i = 0; i < arrA.Length; i++)
            {
                Console.Write("Enter element {0}: ", i);
                arrA[i] = Int32.Parse(Console.ReadLine());
            }

            Console.Write("\nEnter lenght of second array: ");

            if (length != Int32.Parse(Console.ReadLine())) Console.WriteLine("\nArrays have different
lengths.");
            else
            {
                int[] arrB = new int[length];

                for (int i = 0; i < arrB.Length; i++)
                {
                    Console.Write("Enter element {0}: ", i);
                    arrB[i] = Int32.Parse(Console.ReadLine());
                }

                for (int i = 0; i < arrA.Length; i++)
                {
                    if (arrA[i] != arrB[i])
                    {
                        Console.WriteLine("\nArrays are different.");
                        arraysEqual = false;
                        break;
                    }
                }

                if (arraysEqual) Console.WriteLine("\nArrays are the same.");
            }
            }
    }
}
```


Microsoft Visual Studio Debug Console
Enter lenght of first array: 3
Enter element 0: 1
Enter element 1: 2
Enter element 2: 3
Enter lenght of second array: 4
Arrays have different lengths.
C:\Ardit\Universum\Semestri 2\Hyrje në strukturën e të dhënave\Detyra\Chapter 7\ConsoleApp2\ConsoleApp2\bin\Debug\netcoreapp3.1\ConsoleApp2.exe (process 7700) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

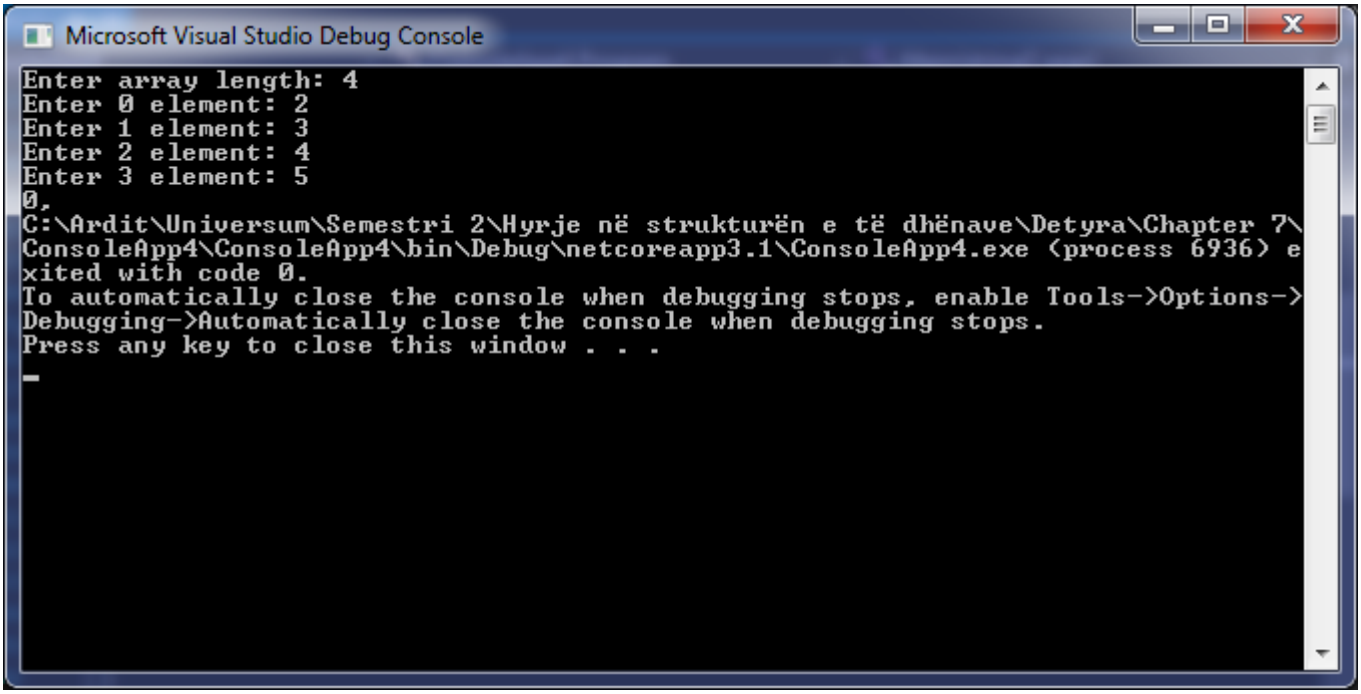
3. Write a program, which **compares two arrays of type char lexicographically** (character by character) and checks, which one is first in the lexicographical order.

```
using System;

namespace ConsoleApp3
{
    class Program
    {
        static void Main(string[] args)
        {
            bool arrayEqual = true;
            char[] arrA = new char[5] { 'a', 'b', 'c', 'd', 'e' };
            char[] arrB = new char[5] { 'a', 'b', 'c', 'd', 'e' };

            if (arrA.Length > arrB.Length) Console.WriteLine("Second array is lexicographicaly
first.");
            else if (arrA.Length < arrB.Length) Console.WriteLine("First array is lexicographicaly
first.");
            else
            {
                for (int i = 0; i < arrA.Length; i++)
                {
                    if (arrA[i] < arrB[i])
                    {
                        Console.WriteLine("First array is lexicographicaly first.");
                        arrayEqual = false;
                        break;
                    }
                    if (arrA[i] > arrB[i])
                    {
                        Console.WriteLine("Second array is lexicographicaly first.");
                        arrayEqual = false;
                        break;
                    }
                }

                if (arrayEqual) Console.WriteLine("Arrays are lexicographicaly equal.");
            }
        }
    }
}
```


Microsoft Visual Studio Debug Console
Arrays are lexicographicaly equal.
C:\Ardit\Universum\Semestri 2\Hyrje në strukturën e të dhënave\Detyra\Chapter 7\ConsoleApp3\ConsoleApp3\bin\Debug\netcoreapp3.1\ConsoleApp3.exe (process 2204) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

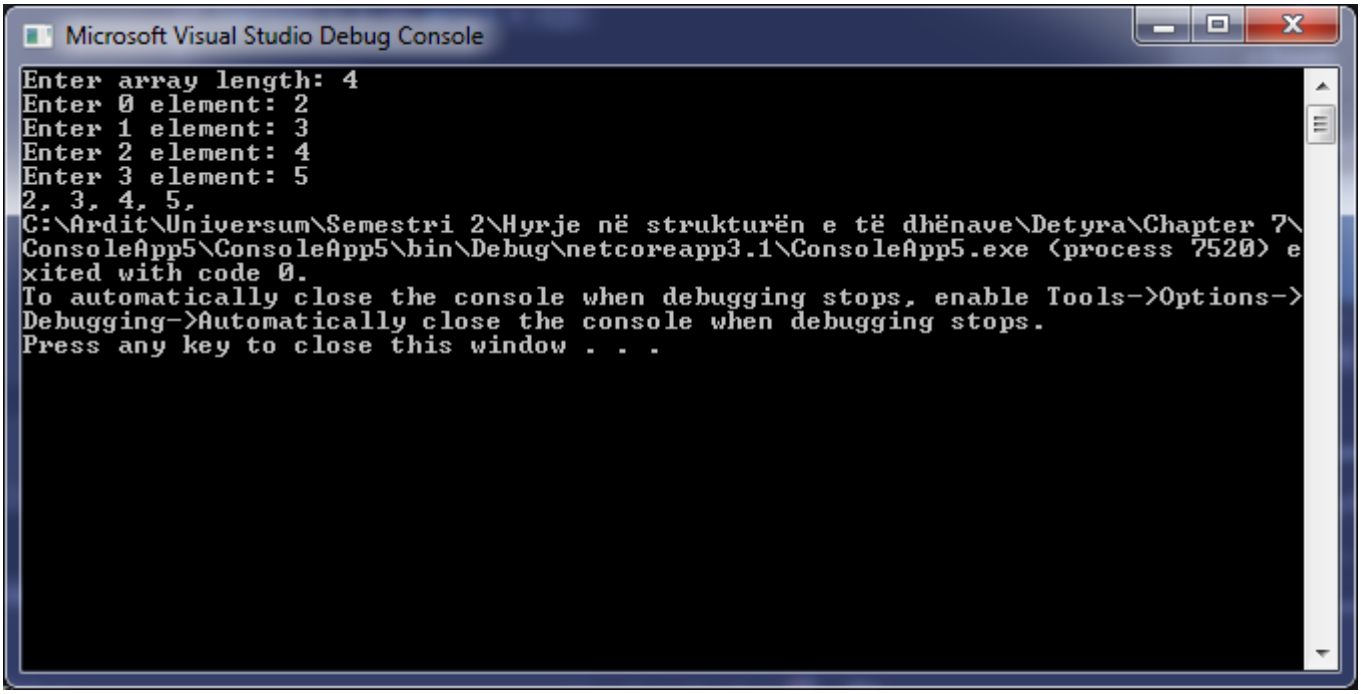
4. Write a program, which finds the maximal sequence of consecutive equal elements in an array.
E.g.: {2, 1, 1, 2, 3, 3, **2, 2, 2**, 1} → {2, 2, 2}.
→ {2, 2, 2}.

```
using System;

namespace ConsoleApp4
{
    class Program
    {
            static void Main(string[] args)
            {
                int count = 1, tempCount = 1, number = 0;

                Console.Write("Enter array length: ");
                int length = Int32.Parse(Console.ReadLine());
                int[] arr = new int[length];

                for (int i = 0; i < arr.Length; i++)
                {
                    Console.Write("Enter {0} element: ", i);
                    arr[i] = Int32.Parse(Console.ReadLine());
                }

                for (int i = 0; i < arr.Length - 1; i++)
                {
                    if (arr[i] == arr[i + 1]) tempCount++;
                    else tempCount = 1;

                    if (tempCount > count)
                    {
                        count = tempCount;
                        number = arr[i];
                    }
                }

                for (int i = 0; i < count; i++) Console.Write("{0}, ", number);
            }
        }
}
```


Microsoft Visual Studio Debug Console
Enter array length: 4
Enter 0 element: 2
Enter 1 element: 3
Enter 2 element: 4
Enter 3 element: 5
0,
C:\Ardit\Universum\Semestri 2\Hyrje në strukturën e të dhënave\Detyra\Chapter 7\ConsoleApp4\ConsoleApp4\bin\Debug\netcoreapp3.1\ConsoleApp4.exe (process 6936) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

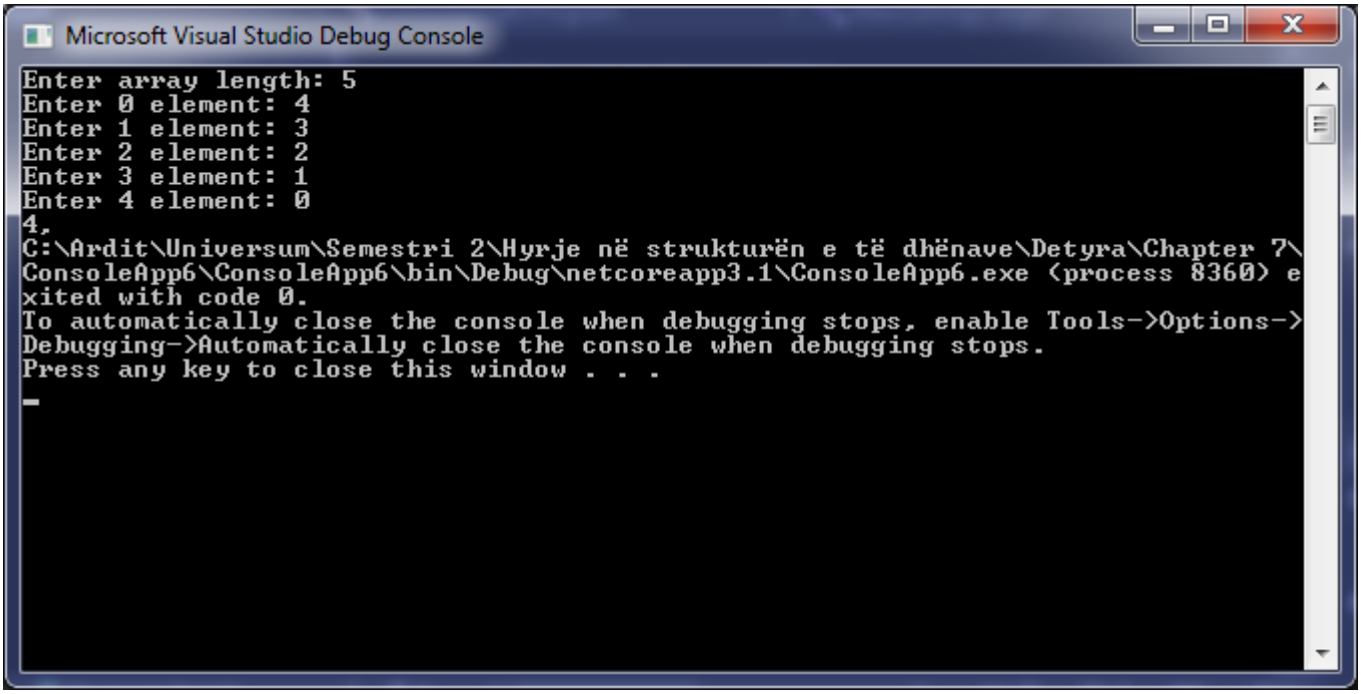
5. Write a program, which finds the **maximal sequence** of consecutively placed **increasing integers**. Example: {3, **2, 3, 4**, 2, 2, 4} → {2, 3, 4}.

```
using System;

namespace ConsoleApp5
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Enter array length: ");
            int length = Int32.Parse(Console.ReadLine());

            int[] arr = new int[length];
            int sames = 1, bestSames = 1, bestStart = 0, lastElement = 0;

            for (int i = 0; i < arr.Length; i++)
            {
                Console.Write("Enter {0} element: ", i);
                arr[i] = Int32.Parse(Console.ReadLine());
            }

            for (int i = 0; i < arr.Length - 1; i++)
            {
                if (arr[i] + 1 == arr[i + 1])
                {
                    sames++;
                    if (sames > bestSames)
                    {
                        bestSames = sames;
                        lastElement = i + 1;
                        bestStart = lastElement - bestSames + 1;
                    }
                }
                else sames = 1;
            }

            for (int i = bestStart; i < bestSames + bestStart; i++) Console.Write("{0}, ", arr[i]);
        }
    }
}
```


Microsoft Visual Studio Debug Console
Enter array length: 4
Enter 0 element: 2
Enter 1 element: 3
Enter 2 element: 4
Enter 3 element: 5
2, 3, 4, 5,
C:\Ardit\Universum\Semestri 2\Hyrje në strukturën e të dhënave\Detyra\Chapter 7\ConsoleApp5\ConsoleApp5\bin\Debug\netcoreapp3.1\ConsoleApp5.exe (process 7520) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

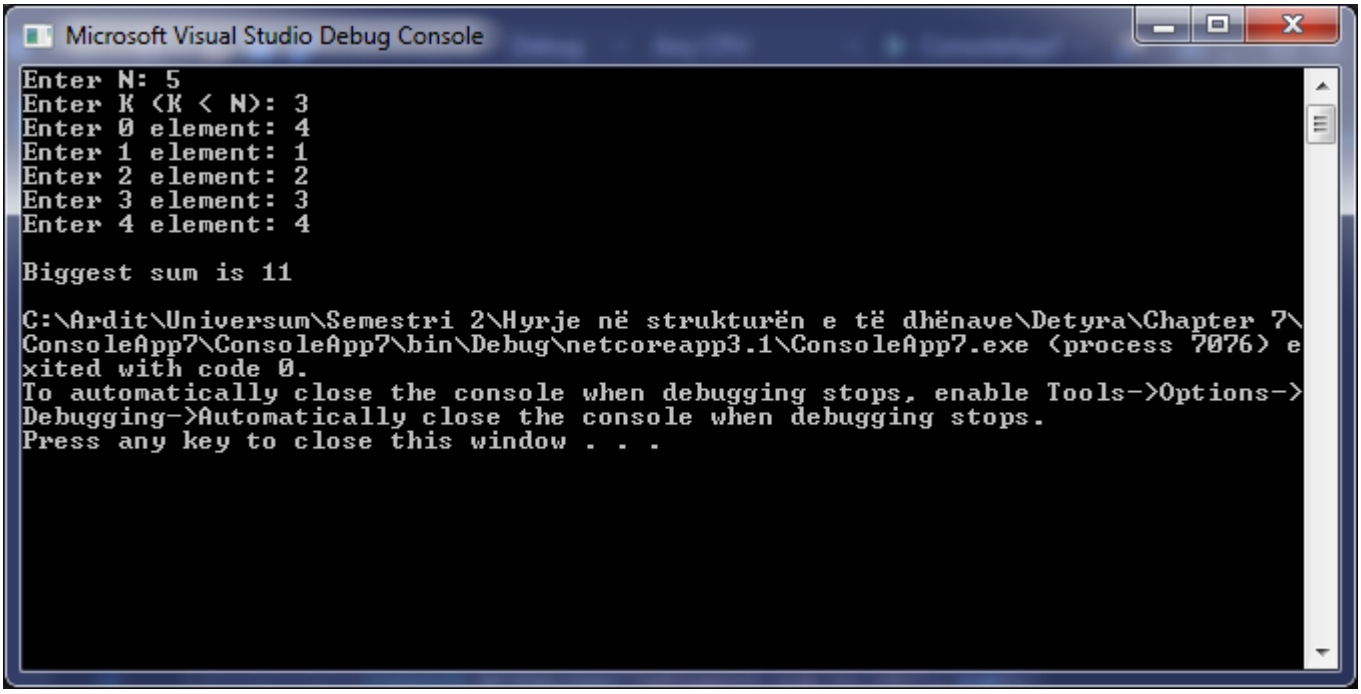
6. Write a program, which finds the **maximal sequence of increasing elements** in an array `arr[n]`. It is not necessary the elements to be consecutively placed.
Example: {9, 6, **2**, 7, **4**, 7, **6**, 5, **8**, 4} → {2, 4, 6, 8}.

```
using System;

namespace ConsoleApp6
{
    class Program
    {
        static void Main(string[] args)
        {
            int counter = 0, tempIndex, tempCounter;

            Console.Write("Enter array length: ");
            int length = Int32.Parse(Console.ReadLine());

            int[] arr = new int[length];
            int[] result = new int[length];

            for (int i = 0; i < length; i++)
            {
                Console.Write("Enter {0} element: ", i);
                arr[i] = Int32.Parse(Console.ReadLine());
            }

            for (int i = 0; i < length; i++)
            {
                int[] tempResult = new int[length];
                tempIndex = tempCounter = 1;
                tempResult[0] = arr[i];

                for (int j = i + 1; j < length; j++)
                {
                    if (arr[j] > tempResult[tempIndex - 1])
                    {
                        tempResult[tempIndex] = arr[j];
                        tempIndex++;
                        tempCounter++;
                    }
                    else if (tempIndex > 1 && arr[j] > tempResult[tempIndex - 2] && arr[j] <
tempResult[tempIndex - 1]) tempResult[tempIndex - 1] = arr[j];
                }

                if (counter < tempCounter)
                {
                    counter = tempCounter;
                    result = tempResult;
                }
            }

            for (int i = 0; i < counter; i++) Console.Write("{0},", result[i]);
        }
    }
}
```


Microsoft Visual Studio Debug Console
Enter array length: 5
Enter 0 element: 4
Enter 1 element: 3
Enter 2 element: 2
Enter 3 element: 1
Enter 4 element: 0
4,
C:\Ardit\Universum\Semestri 2\Hyrje në strukturën e të dhënave\Detyra\Chapter 7\ConsoleApp6\ConsoleApp6\bin\Debug\netcoreapp3.1\ConsoleApp6.exe (process 8360) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

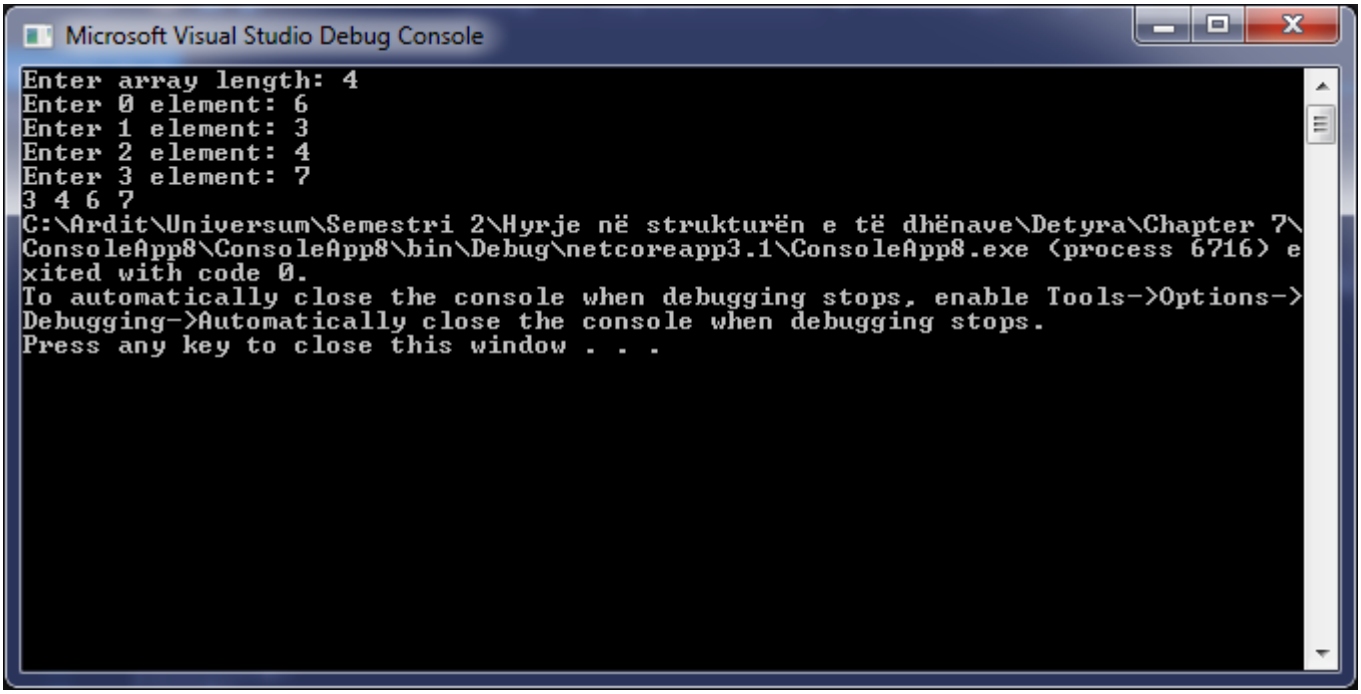
7. Write a program, which reads from the console two integer numbers **N** and **K** (K < N) and array of N integers. Find those **K consecutive elements** in the array, which have **maximal sum**.

```
using System;

namespace ConsoleApp7
{
    class Program
    {
        static void Main(string[] args)
        {
            int sum = 0;

            Console.Write("Enter N: ");
            int n = Int32.Parse(Console.ReadLine());
            Console.Write("Enter K (K < N): ");
            int k = Int32.Parse(Console.ReadLine());

            int[] arr = new int[n];

            for (int i = 0; i < n; i++)
            {
                Console.Write("Enter {0} element: ", i);
                arr[i] = Int32.Parse(Console.ReadLine());
            }

            Array.Sort(arr, (a, b) => b.CompareTo(a));

            for (int i = 0; i < k; i++) sum += arr[i];

            Console.WriteLine("\nBiggest sum is {0}", sum);
        }
    }
}
```



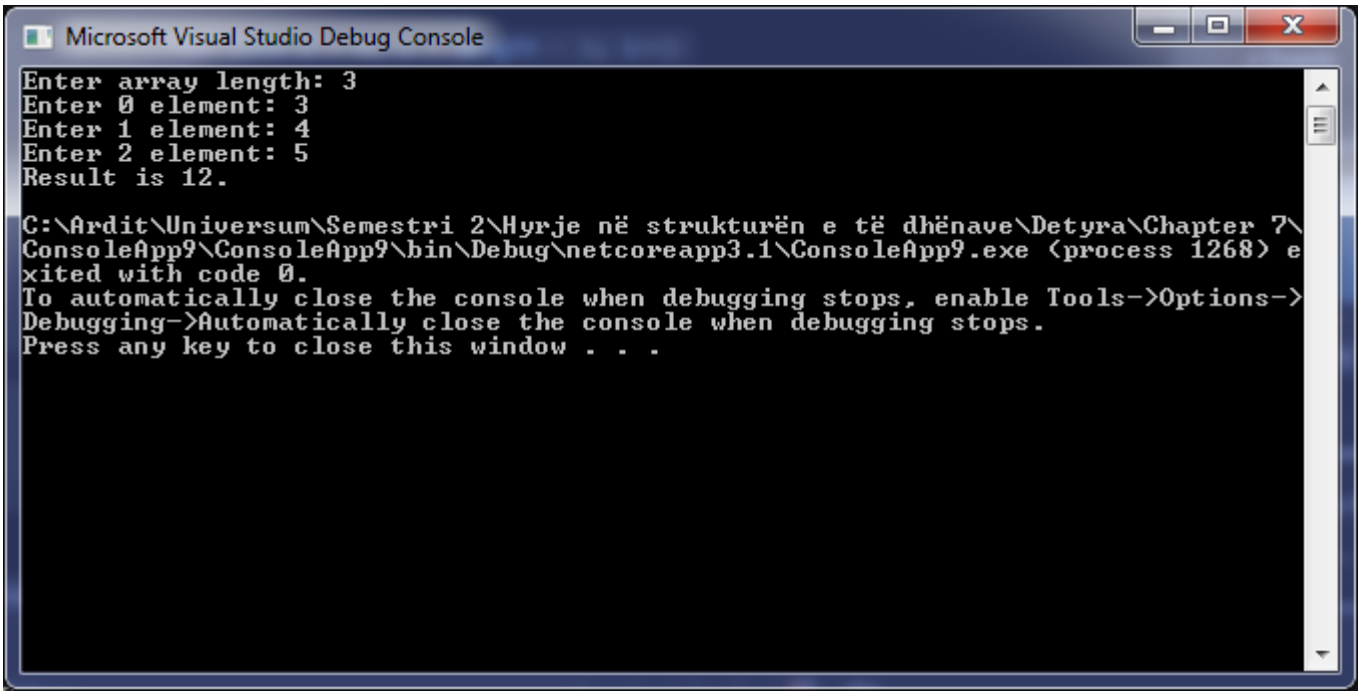
8. **Sorting an array** means to arrange its elements in an increasing (or decreasing) order. Write a program, which sorts an array using the algorithm "`selection sort`".

```
using System;

namespace ConsoleApp8
{
    class Program
    {
        static void Main(string[] args)
        {

                int i, j, iMin, temp;

                Console.Write("Enter array length: ");
                int length = Int32.Parse(Console.ReadLine());

                int[] arr = new int[length];

                for (i = 0; i < length; i++)
                {
                    Console.Write("Enter {0} element: ", i);
                    arr[i] = Int32.Parse(Console.ReadLine());
                }

                for (j = 0; j < length - 1; j++)
                {
                    iMin = j;

                    for (i = j + 1; i < length; i++) if (arr[i] < arr[iMin]) iMin = i;

                    if (iMin != j)
                    {
                        temp = arr[j];
                        arr[j] = arr[iMin];
                        arr[iMin] = temp;
                    }
                }

                for (i = 0; i < length; i++) Console.Write("{0} ", arr[i]);
            }
        }
    }
```


Microsoft Visual Studio Debug Console
Enter array length: 4
Enter 0 element: 6
Enter 1 element: 3
Enter 2 element: 4
Enter 3 element: 7
3 4 6 7
C:\Ardit\Universum\Semestri 2\Hyrje në strukturën e të dhënave\Detyra\Chapter 7\ConsoleApp8\ConsoleApp8\bin\Debug\netcoreapp3.1\ConsoleApp8.exe (process 6716) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

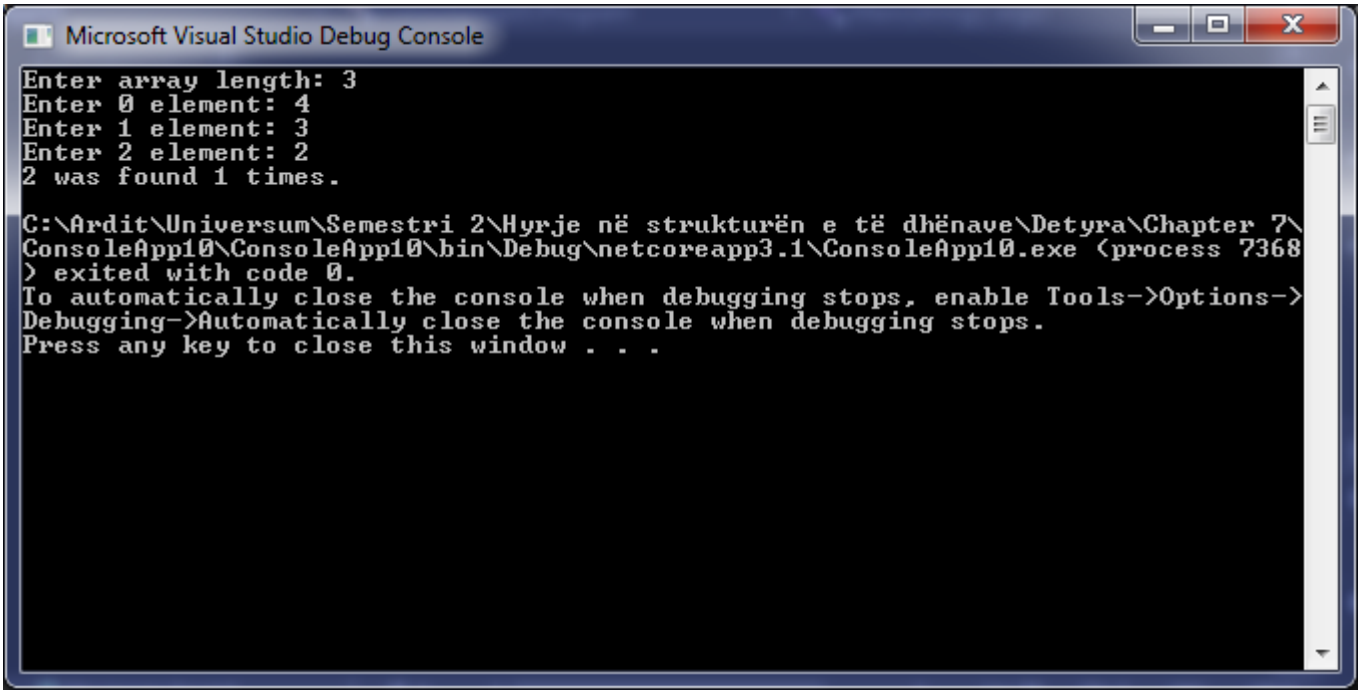
9. Write a program, which finds a subsequence of numbers with maximal sum. E.g.: {2, 3, -6, -1, **2, -1, 6, 4**, -8, 8} → **11**

```
using System;

namespace ConsoleApp9
{
    class Program
    {
            static void Main(string[] args)
            {
                int sum = 0, tempSum;

                Console.Write("Enter array length: ");
                int length = Int32.Parse(Console.ReadLine());

                int[] arr = new int[length];

                for (int i = 0; i < length; i++)
                {
                    Console.Write("Enter {0} element: ", i);
                    arr[i] = Int32.Parse(Console.ReadLine());
                }

                for (int i = 0; i < length - 1; i++)
                {
                    tempSum = arr[i];

                    for (int j = i + 1; j < length; j++)
                    {
                        tempSum += arr[j];
                        if (tempSum > sum) sum = tempSum;
                    }
                }

                Console.WriteLine("Result is {0}. ", sum);
            }
        }
}
```



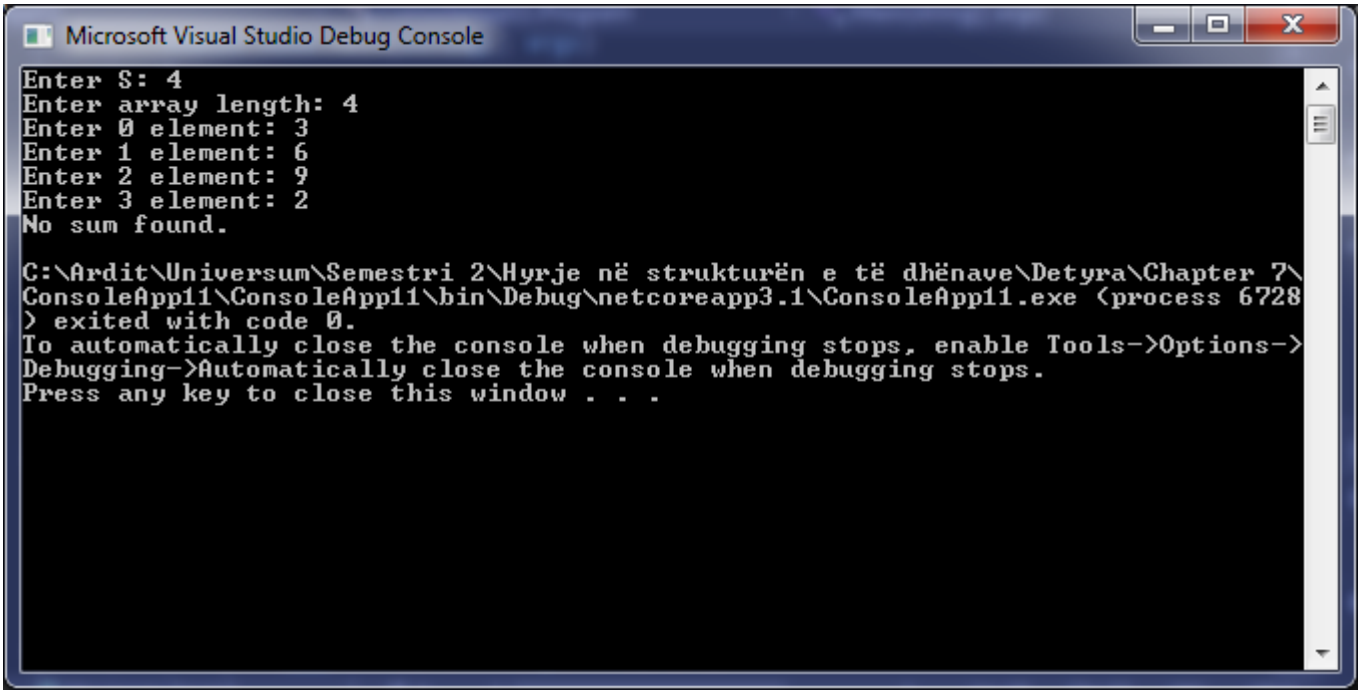
10. Write a program, which finds the **most frequently occurring** element in an array. Example: {**4**, 1, 1, **4**, 2, 3, **4**, **4**, 1, 2, **4**, 9, 3} → 4 (5 times).

```
using System;

namespace ConsoleApp10
{
    class Program
    {
        static void Main(string[] args)
        {
            int counter = 0, tempCounter = 1, foundNumber = 0;

            Console.Write("Enter array length: ");
            int length = Int32.Parse(Console.ReadLine());

            int[] arr = new int[length];

            for (int i = 0; i < length; i++)
            {
                Console.Write("Enter {0} element: ", i);
                arr[i] = Int32.Parse(Console.ReadLine());
            }

            Array.Sort(arr);

            for (int i = 0; i < length - 1; i++)
            {
                if (arr[i] == arr[i + 1]) tempCounter++;
                else tempCounter = 1;
                if (tempCounter > counter)
                {
                    counter = tempCounter;
                    foundNumber = arr[i];
                }
            }

            Console.WriteLine("{0} was found {1} times.", foundNumber, counter);
        }
    }
}
```


Microsoft Visual Studio Debug Console
Enter array length: 3
Enter 0 element: 4
Enter 1 element: 3
Enter 2 element: 2
2 was found 1 times.
C:\Ardit\Universum\Semestri 2\Hyrje në strukturën e të dhënave\Detyra\Chapter 7\ConsoleApp10\ConsoleApp10\bin\Debug\netcoreapp3.1\ConsoleApp10.exe (process 7368) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

11. Write a program to find a sequence of neighbor numbers in an array, which has a **sum of certain number S**. Example: {4, 3, 1, **4**, **2**, **5**, 8}, S=11 → {4, 2, 5}.

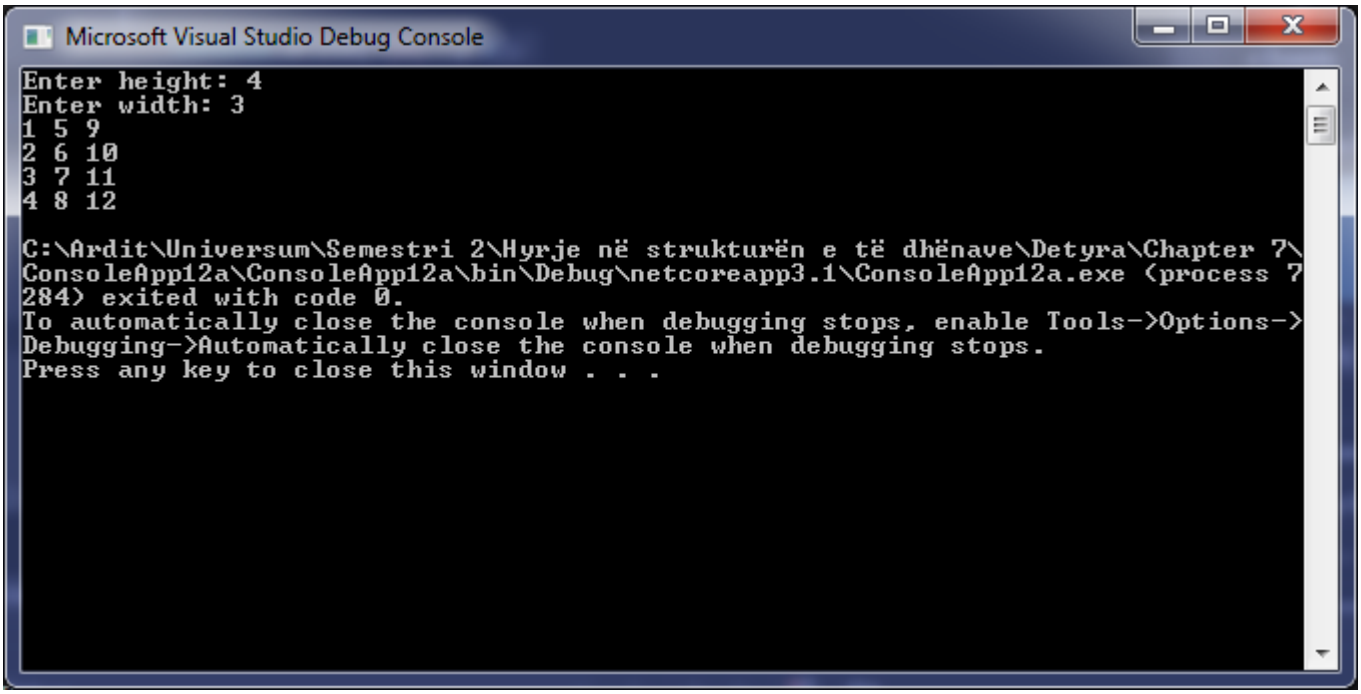
```
using System;

namespace ConsoleApp11
{
    class Program
    {
        static void Main(string[] args)
        {
            int sum = 0, start = 0, end = 0;
            bool sumFound = false;

            Console.Write("Enter S: ");
            int s = Int32.Parse(Console.ReadLine());

            Console.Write("Enter array length: ");
            int length = Int32.Parse(Console.ReadLine());

            int[] arr = new int[length];

            for (int i = 0; i < length; i++)
            {
                Console.Write("Enter {0} element: ", i);
                arr[i] = Int32.Parse(Console.ReadLine());
            }

            for (int i = 0; i < length - 1; i++)
            {
                sum = arr[i];

                for (int j = i + 1; j < length; j++)
                {
                    sum += arr[j];
                    if (sum == s)
                    {
                        start = i;
                        end = j;
                        sumFound = true;
                        break;
                    }
                }

                if (sumFound) break;
            }

            if (sumFound) for (int i = start; i <= end; i++) Console.Write("{0},", arr[i]);
            else Console.WriteLine("No sum found.");
        }
    }
}
```


Microsoft Visual Studio Debug Console
Enter S: 4
Enter array length: 4
Enter 0 element: 3
Enter 1 element: 6
Enter 2 element: 9
Enter 3 element: 2
No sum found.
C:\Ardit\Universum\Semestri 2\Hyrje në strukturën e të dhënave\Detyra\Chapter 7\ConsoleApp11\ConsoleApp11\bin\Debug\netcoreapp3.1\ConsoleApp11.exe (process 6728) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

12. Write a program that creates the following square matrices and displays them on the console in formatted form. The size of the matrices is entered from the console.

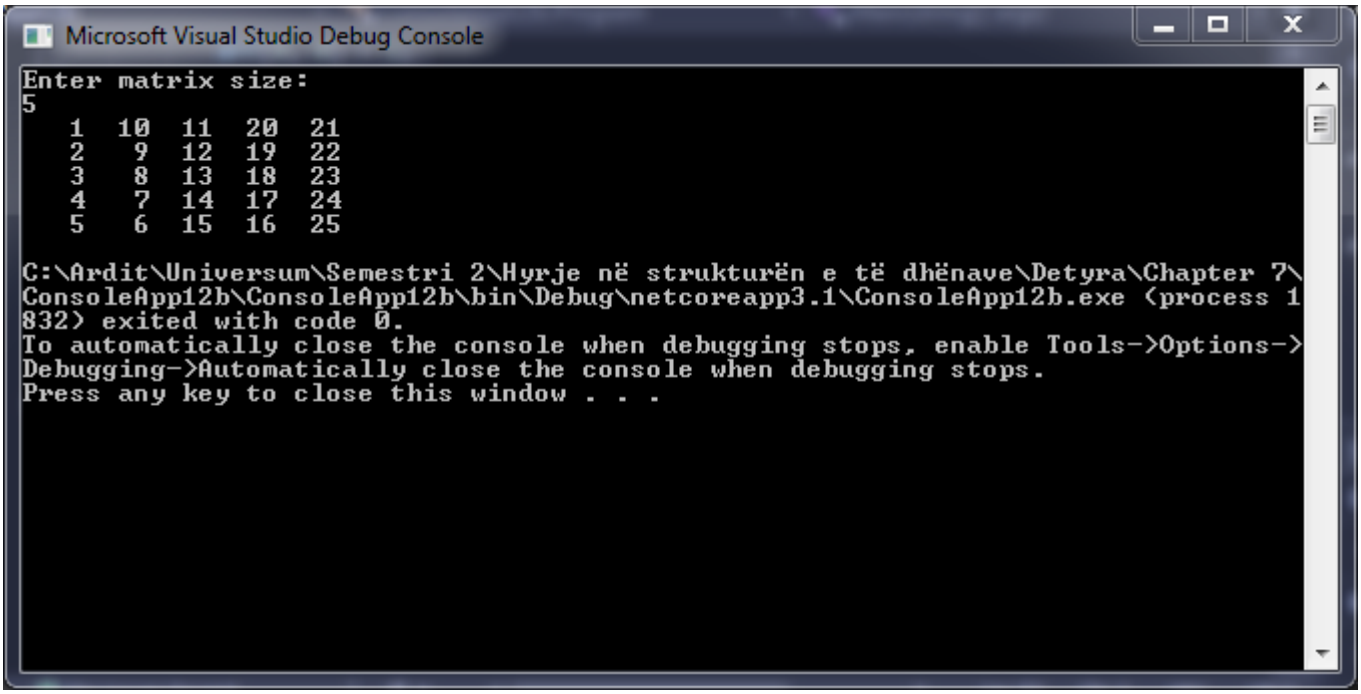
Example for (4.4):

```
using System;

namespace ConsoleApp12a
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Enter height: ");
            int y = Int32.Parse(Console.ReadLine());

            Console.Write("Enter width: ");
            int x = Int32.Parse(Console.ReadLine());

            int a = 0;

            for (int i = 1; i <= y; i++)
            {
                Console.Write("{0} ", i);

                a += i;

                for (int j = 1; j < x; j++)
                {
                    a += y;
                    Console.Write("{0} ", a);
                }

                a = 0;
                Console.WriteLine();
            }
        }
    }
}
```



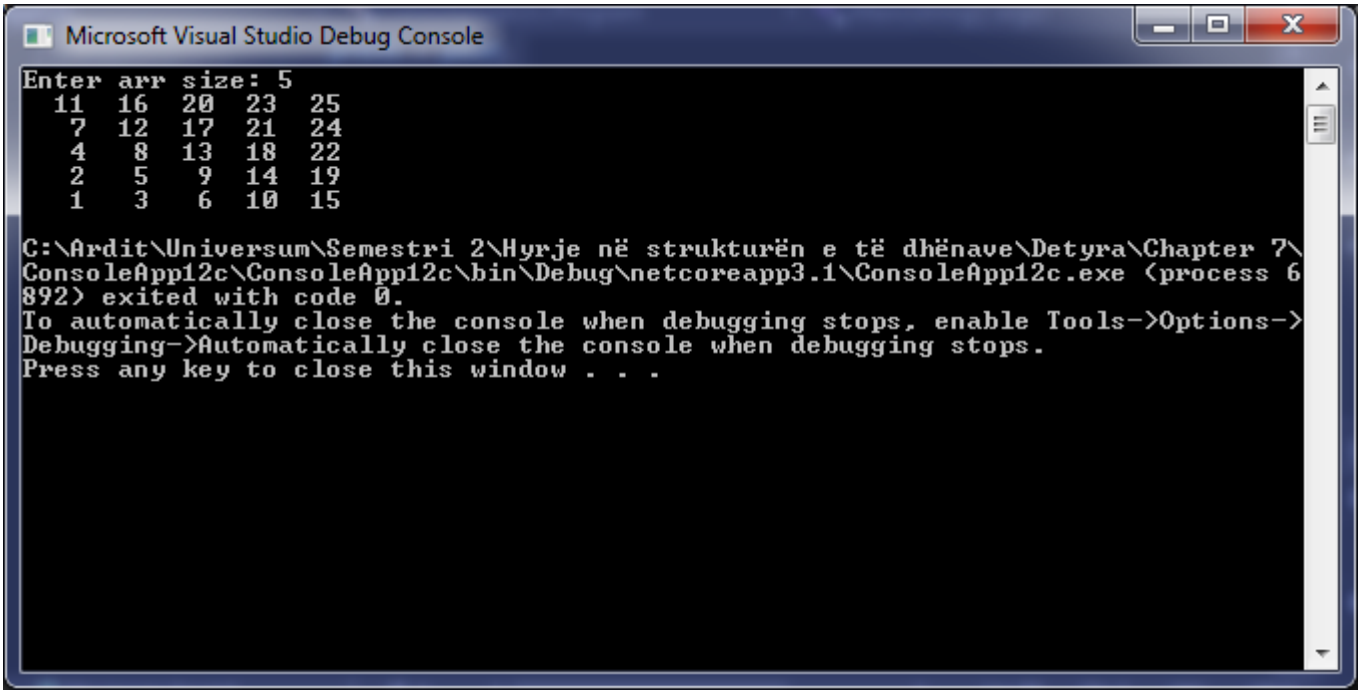
12 b

```
using System;

namespace ConsoleApp12b
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Enter matrix size: ");
            int n = Int32.Parse(Console.ReadLine());

            int[,] arr = new int[n, n];
            arr[0, 0] = 1;

            for (int i = 1; i < arr.GetLength(0); i++)
                if (i % 2 == 1) arr[0, i] = arr[0, i - 1] + n * 2 - 1;
                else arr[0, i] = arr[0, i - 1] + 1;

            for (int i = 1; i < arr.GetLength(0); i++)
                for (int j = 0; j < arr.GetLength(1); j++)
                    if (j % 2 == 1) arr[i, j] = arr[i - 1, j] - 1;
                    else arr[i, j] = arr[i - 1, j] + 1;

            for (int i = 0; i < arr.GetLength(0); i++)
            {
                for (int j = 0; j < arr.GetLength(1); j++) Console.Write("{0, 4}", arr[i, j]);
                Console.WriteLine();
            }
        }
    }
}
```



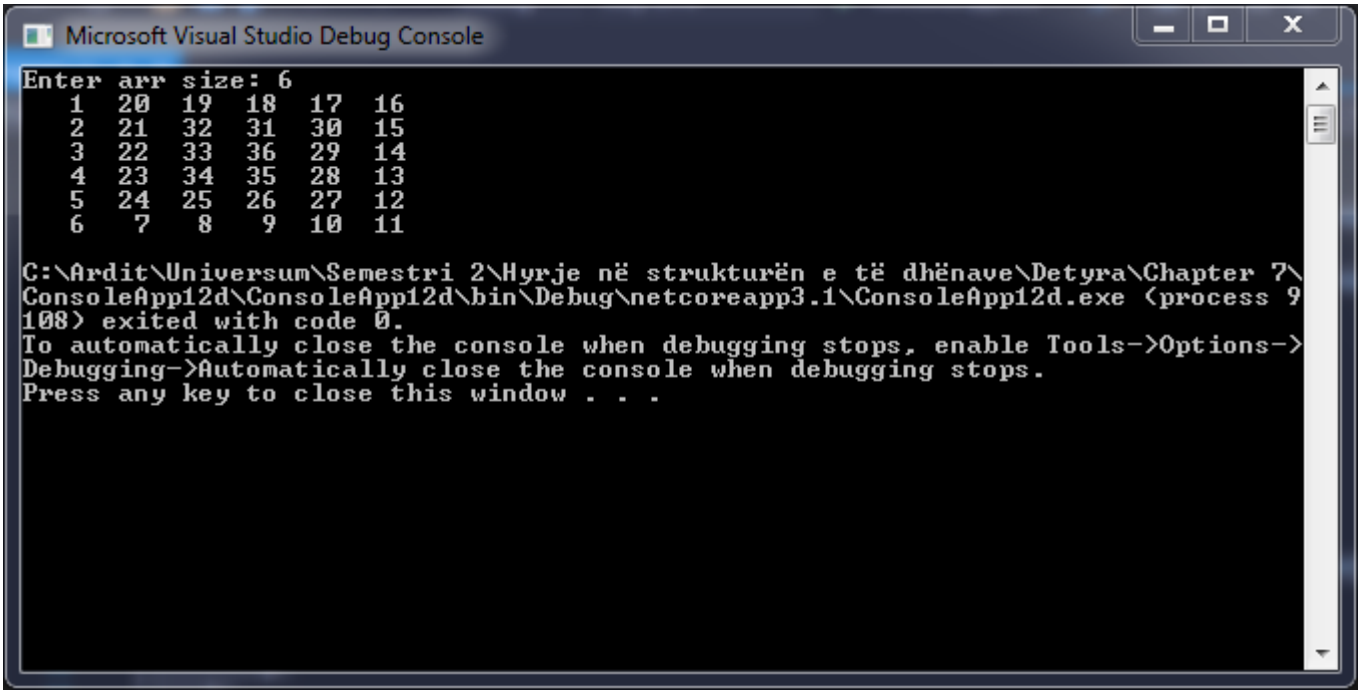
12 c

```
using System;

namespace ConsoleApp12c
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Enter arr size: ");
            int length = Int32.Parse(Console.ReadLine());

            int[,] arr = new int[length, length];
            arr[length - 1, 0] = 1;
            int counter = 1;
            for (int row = length - 2; row >= 0; row--)
            {
                arr[row, 0] = arr[row + 1, 0] + counter;
                int newRow = row;
                for (int diagonal = 1; diagonal <= counter; diagonal++)
                {
                    arr[newRow + 1, diagonal] = arr[newRow, diagonal - 1] + 1;
                    newRow++;
                }
                counter++;
            }

            arr[0, length - 1] = length * length;
            int diagonalLength = 2;
            int posX = 1;
            int posY = length - 1;
            int prevX = 0;
            int prevY = length - 1;

            for (int i = 1; i < counter - 1; i++)
            {
                for (int j = 1; j <= diagonalLength; j++)
                {
                    arr[posX, posY] = arr[prevX, prevY] - 1;
                    prevX = posX;
                    prevY = posY;
                    posX--;
                    posY--;
                }
                diagonalLength++;
                posX = i + 1;
                posY = length - 1;
            }

            for (int i = 0; i < arr.GetLength(0); i++)
            {
                for (int j = 0; j < arr.GetLength(1); j++) Console.Write("{0, 4}", arr[i, j]);
                Console.WriteLine();
            }

        }
    }
}
```


Microsoft Visual Studio Debug Console
Enter arr size: 5
11 16 20 23 25
7 12 17 21 24
4 8 13 18 22
2 5 9 14 19
1 3 6 10 15
C:\Ardit\Universum\Semestri 2\Hyrje në strukturën e të dhënave\Detyra\Chapter 7\ConsoleApp12c\ConsoleApp12c\bin\Debug\netcoreapp3.1\ConsoleApp12c.exe (process 6892) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

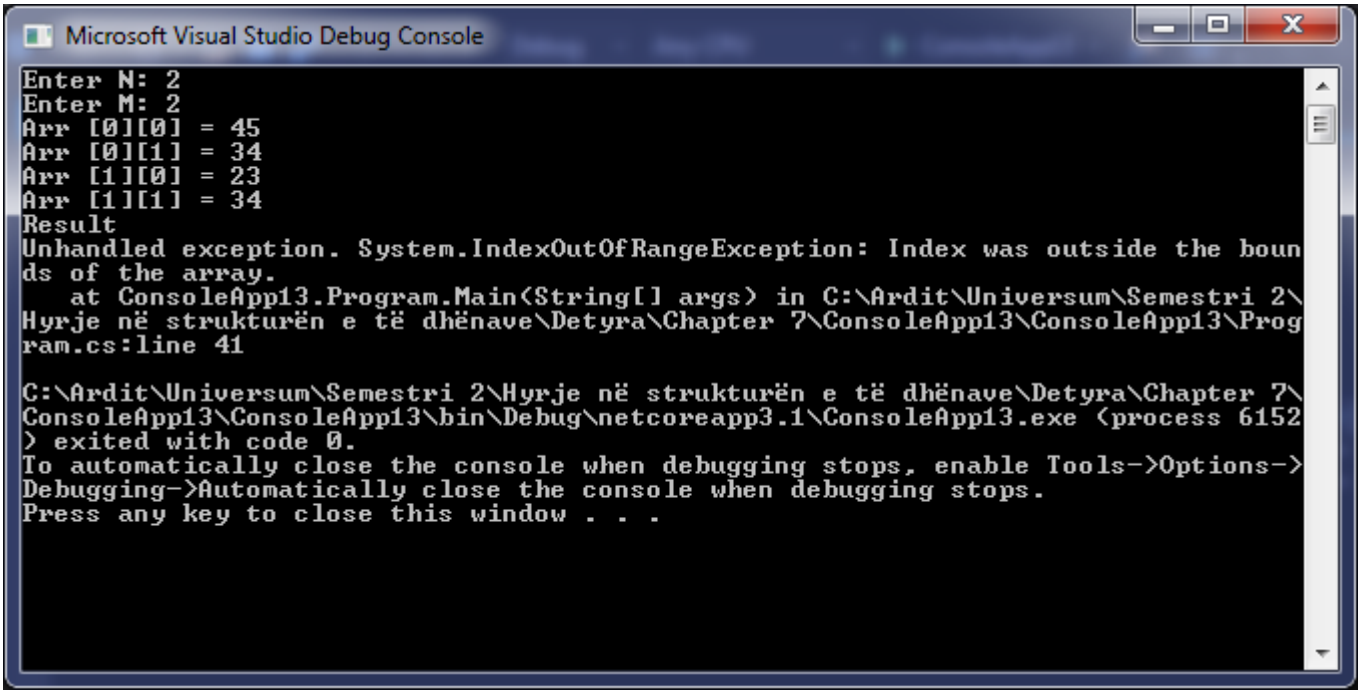
12 d

```
using System;

namespace ConsoleApp12d
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Enter arr size: ");
            int length = Int32.Parse(Console.ReadLine());

            int[,] arr = new int[length, length];
            int numConcentricSquares = (int)Math.Ceiling((length) / 2.0);
            int sideLen = length;
            int currNum = 0;

            for (int i = 0; i < numConcentricSquares; i++)
            {

                for (int j = 0; j < sideLen; j++) arr[i + j, i] = ++currNum;

                for (int j = 1; j < sideLen - 1; j++) arr[length - 1 - i, i + j] = ++currNum;

                for (int j = sideLen - 1; j > 0; j--) arr[i + j, length - 1 - i] = ++currNum;

                for (int j = sideLen - 1; j > 0; j--) arr[i, i + j] = ++currNum;

                sideLen -= 2;
            }

            for (int i = 0; i < arr.GetLength(0); i++)
            {
                for (int j = 0; j < arr.GetLength(1); j++) Console.Write("{0, 4}", arr[i, j]);
                Console.WriteLine();
            }
        }
    }
}
```


Microsoft Visual Studio Debug Console
Enter arr size: 6
1 20 19 18 17 16
2 21 32 31 30 15
3 22 33 36 29 14
4 23 34 35 28 13
5 24 25 26 27 12
6 7 8 9 10 11
C:\Ardit\Universum\Semestri 2\Hyrje në strukturën e të dhënave\Detyra\Chapter 7\ConsoleApp12d\ConsoleApp12d\bin\Debug\netcoreapp3.1\ConsoleApp12d.exe (process 9108) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

13. Write a program, which creates a rectangular array with size of `n` by `m` elements. The dimensions and the elements should be read from the console. Find a **platform with size of (3, 3) with a maximal sum**.

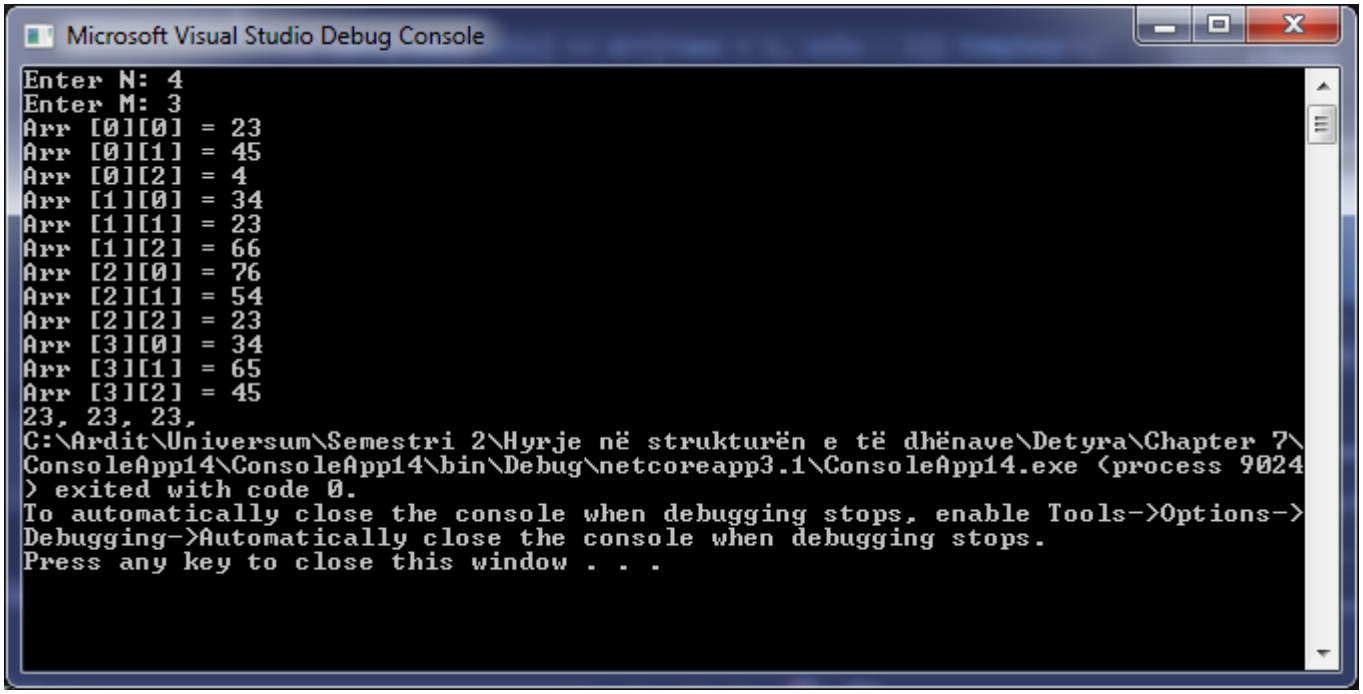
```
using System;

namespace ConsoleApp13
{
    class Program
    {
        static void Main(string[] args)
        {
            int row = 0, col = 0, sum = -1000;

            Console.Write("Enter N: ");
            int n = Int32.Parse(Console.ReadLine());
            Console.Write("Enter M: ");
            int m = Int32.Parse(Console.ReadLine());

            int[,] arr = new int[n, m];

            for (int i = 0; i < n; i++)
                for (int j = 0; j < m; j++)
                {
                    Console.Write("Arr [{0}][{1}] = ", i, j);
                    arr[i, j] = Int32.Parse(Console.ReadLine());
                }

            for (int tempRow = 0; tempRow < arr.Length - 2; tempRow++)
                for (int tempCol = 0; tempCol < arr.GetLength(0) - 2; tempCol++)
                {
                    int tempSum = arr[row, col] + arr[row, col + 1] + arr[row, col + 2] +
                        arr[row + 1, col] + arr[row + 1, col + 1] + arr[row + 1, col + 2] +
                        arr[row + 2, col] + arr[row + 2, col + 1] + arr[row + 2, col + 2];

                    if (tempSum > sum)
                    {
                        row = tempRow;
                        col = tempCol;
                        sum = tempSum;
                    }
                }

            Console.WriteLine("Result");
            Console.WriteLine("{0} {1} {2}", arr[row, col], arr[row, col + 1], arr[row, col + 2]);
            Console.WriteLine("{0} {1} {2}", arr[row + 1, col], arr[row + 1, col + 1], arr[row + 1, col
+ 2]);
            Console.WriteLine("{0} {1} {2}", arr[row + 2, col], arr[row + 2, col + 2], arr[row + 2, col
+ 2]);
            Console.WriteLine("The maximum sum is {0}.", sum);
        }
    }
}
```


Microsoft Visual Studio Debug Console
Enter N: 2
Enter M: 2
Arr [0][0] = 45
Arr [0][1] = 34
Arr [1][0] = 23
Arr [1][1] = 34
Result
Unhandled exception. System.IndexOutOfRangeException: Index was outside the boun
ds of the array.
at ConsoleApp13.Program.Main(String[] args) in C:\Ardit\Universum\Semestri 2\
Hyrje në strukturën e të dhënave\Detyra\Chapter 7\ConsoleApp13\ConsoleApp13\Prog
ram.cs:line 41
C:\Ardit\Universum\Semestri 2\Hyrje në strukturën e të dhënave\Detyra\Chapter 7\
ConsoleApp13\ConsoleApp13\bin\Debug\netcoreapp3.1\ConsoleApp13.exe (process 6152
) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->
Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

14. Write a program, which finds the **longest sequence of equal string elements in a matrix**. A sequence in a matrix we define as a set of neighbor elements **on the same row, column or diagonal**.

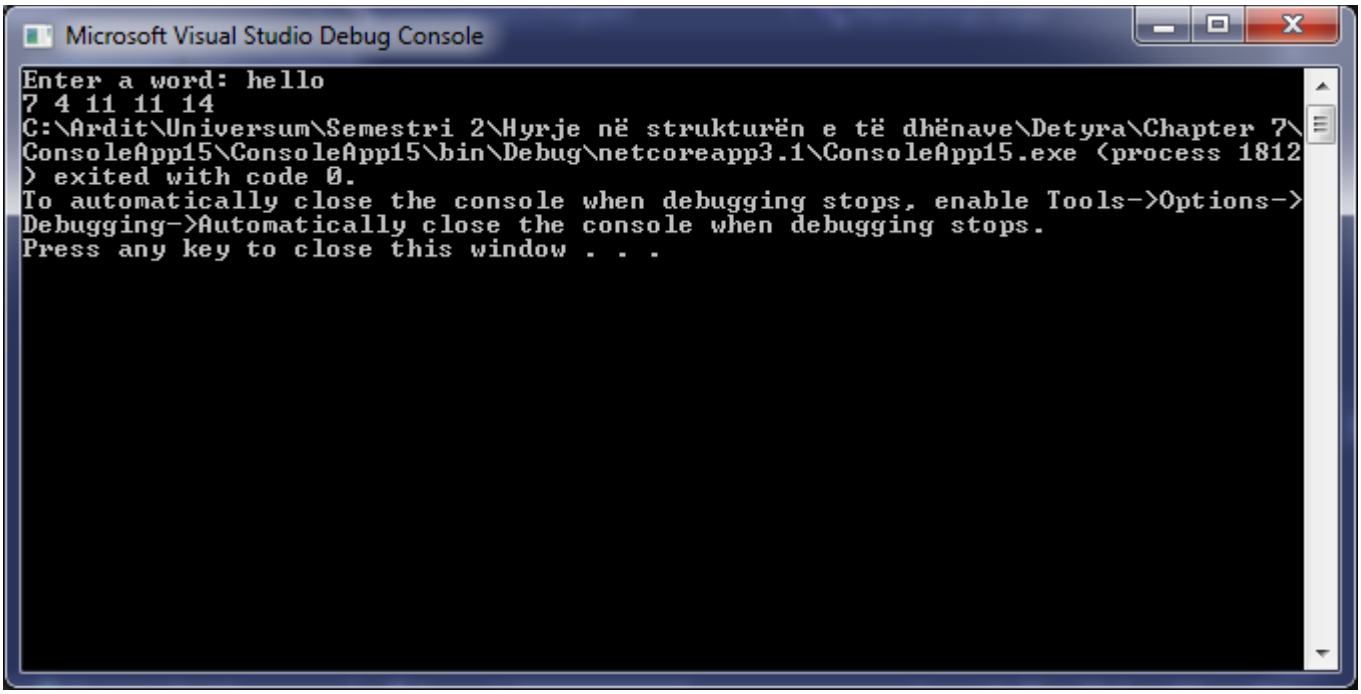
```
using System;

namespace ConsoleApp14
{
    class Program
    {
        static void Main(string[] args)
        {
            int tempSeq = 1, seq = 1;
            string element = "e";

            Console.Write("Enter N: ");
            int n = Int32.Parse(Console.ReadLine());
            Console.Write("Enter M: ");
            int m = Int32.Parse(Console.ReadLine());

            string[,] arr = new string[n, m];

            for (int i = 0; i < n; i++)
                for (int j = 0; j < m; j++)
                {
                    Console.Write("Arr [{0}][{1}] = ", i, j);
                    arr[i, j] = Console.ReadLine();
                }


            for (int rows = 0; rows < arr.GetLength(0); rows++)
            {
                for (int cols = 0; cols < arr.GetLength(1) - 1; cols++)
                {
                    if (arr[rows, cols] == arr[rows, cols + 1]) tempSeq++;
                    else tempSeq = 1;

                    if (seq < tempSeq)
                    {
                        seq = tempSeq;
                        element = arr[rows, cols];
                    }
                }
                tempSeq = 1;
            }

            for (int cols = 0; cols < arr.GetLength(1); cols++)
            {
                for (int rows = 0; rows < arr.GetLength(0) - 1; rows++)
                {
                    if (arr[rows, cols] == arr[rows + 1, cols]) tempSeq++;
                    else tempSeq = 1;

                    if (seq < tempSeq)
                    {
                        seq = tempSeq;
                        element = arr[rows, cols];
                    }
                }
                tempSeq = 1;
            }
```

```
            for (int i = 0; i < arr.GetLength(0) - 1; i++)
                for (int j = 0; j < arr.GetLength(1) - 1; j++)
                {
                    for (int rows = i, cols = j; rows < arr.GetLength(0) - 1 && cols < arr.GetLength(1)
- 1; rows++, cols++)
                    {
                        if (arr[rows, cols] == arr[rows + 1, cols + 1]) tempSeq++;
                        else tempSeq = 1;

                        if (seq < tempSeq)
                        {
                            seq = tempSeq;
                            element = arr[rows, cols];
                        }
                    }
                    tempSeq = 1;
                }

            for (int i = 0; i < arr.GetLength(0) - 1; i++)
                for (int j = 1; j < arr.GetLength(1); j++)
                {
                    for (int rows = i, cols = j; rows < arr.GetLength(0) - 1 && cols > 0; rows++, cols-
-)
                    {
                        if (arr[rows, cols] == arr[rows + 1, cols - 1]) tempSeq++;
                        else tempSeq = 1;

                        if (seq < tempSeq)
                        {
                            seq = tempSeq;
                            element = arr[rows, cols];
                        }
                    }
                    tempSeq = 1;
                }

            for (int i = 0; i < seq; i++) Console.Write("{0}, ", element);
        }
    }
}
```


Microsoft Visual Studio Debug Console
Enter N: 4
Enter M: 3
Arr [0][0] = 23
Arr [0][1] = 45
Arr [0][2] = 4
Arr [1][0] = 34
Arr [1][1] = 23
Arr [1][2] = 66
Arr [2][0] = 76
Arr [2][1] = 54
Arr [2][2] = 23
Arr [3][0] = 34
Arr [3][1] = 65
Arr [3][2] = 45
23, 23, 23,
C:\Ardit\Universum\Semestri 2\Hyrje në strukturën e të dhënave\Detyra\Chapter 7\ConsoleApp14\ConsoleApp14\bin\Debug\netcoreapp3.1\ConsoleApp14.exe (process 9024) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

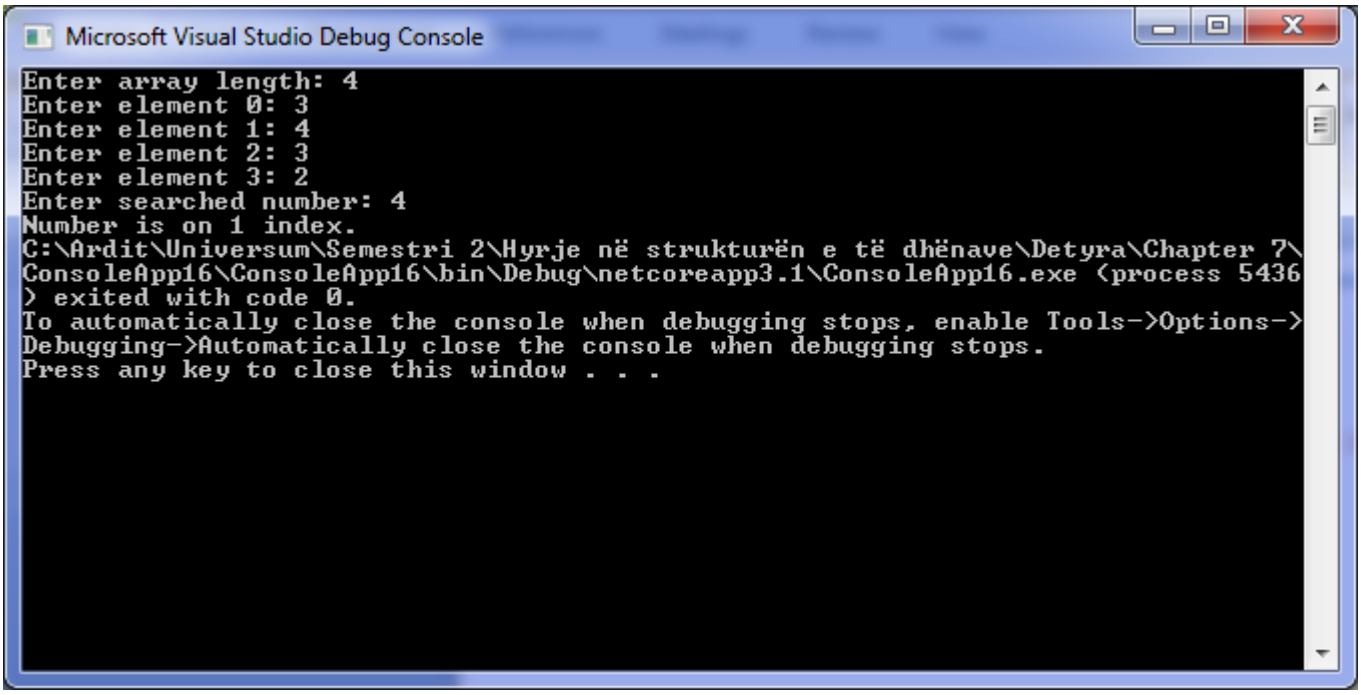
15. Write a program, which creates an array containing **all Latin letters**. The user inputs **a word** from the console and as result the program prints to the console the **indices of the letters from the word**.

```
using System;

namespace ConsoleApp15
{
    class Program
    {
        static void Main(string[] args)
        {
            char[] alphabet = { 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n',
'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z' };

            Console.Write("Enter a word: ");
            char[] word = (Console.ReadLine()).ToCharArray();

            for (int i = 0; i < word.Length; i++)
                for (int j = 0; j < alphabet.Length; j++)
                    if (word[i] == alphabet[j]) Console.Write("{0} ", j);
        }
    }
}
```



16. Write a program, which uses a **binary search** in a **sorted** array of integer numbers to find a certain element.

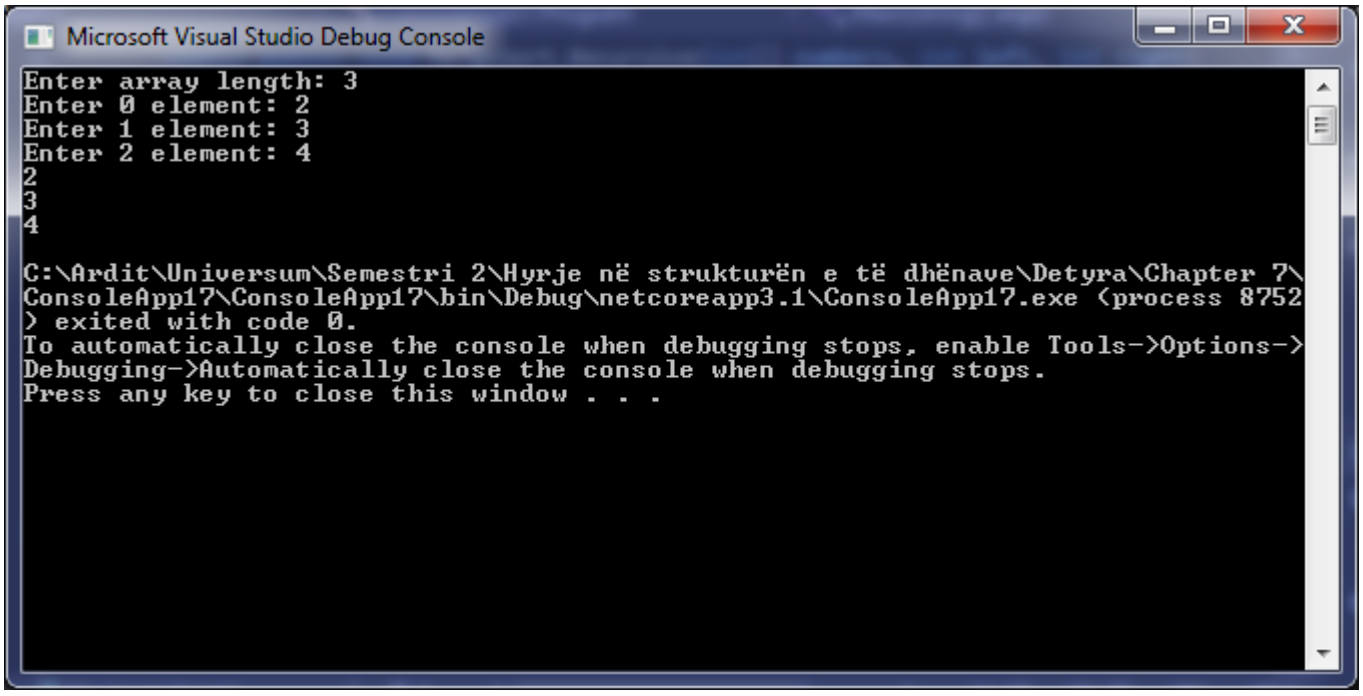
```
using System;

namespace ConsoleApp16
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Enter array length: ");
            int length = Int32.Parse(Console.ReadLine());

            int[] arr = new int[length];

            for (int i = 0; i < length; i++)
            {
                Console.Write("Enter element {0}: ", i);
                arr[i] = Int32.Parse(Console.ReadLine());
            }

            Console.Write("Enter searched number: ");
            int number = Int32.Parse(Console.ReadLine());

            int index = Array.BinarySearch(arr, number);

            if (index >= 0) Console.Write("Number is on {0} index.", index);
            else Console.Write("Number wasn't found.");
        }
    }
}
```



17. Write a program, which sorts an array of integer elements using a "**merge sort**" algorithm.

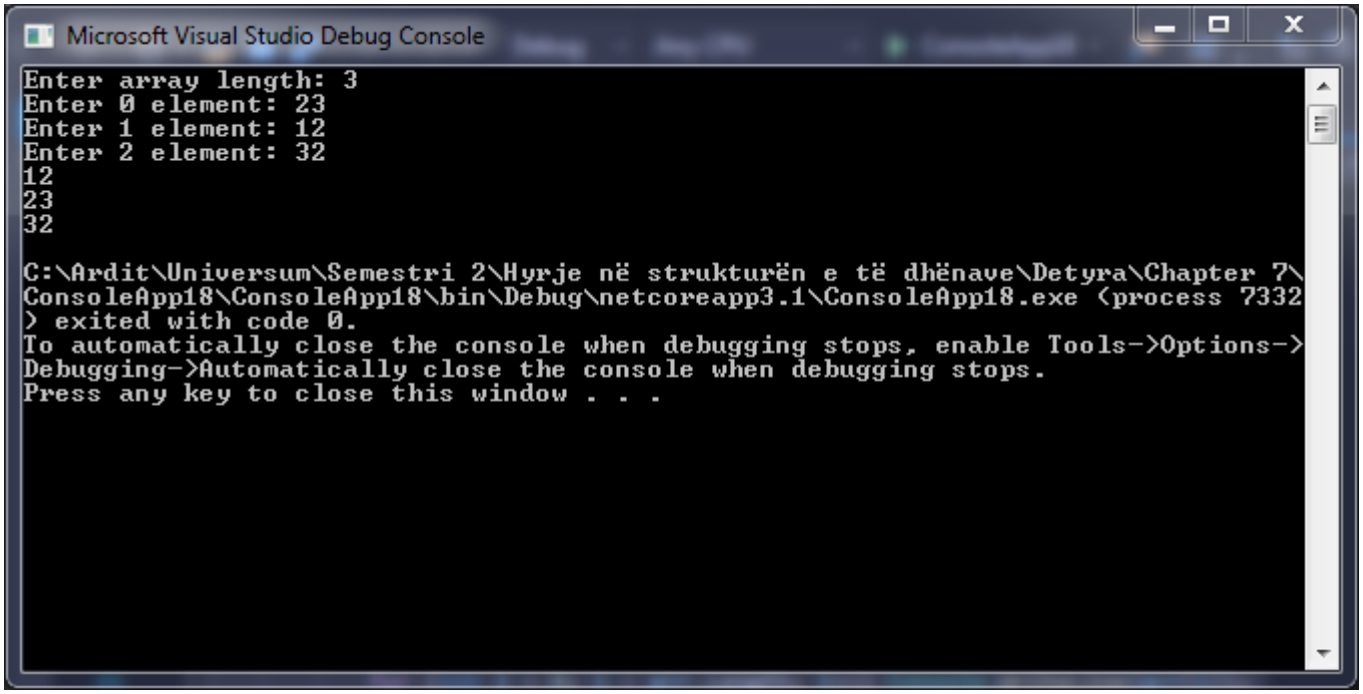
```
using System;

namespace ConsoleApp17
{
    class Program
    {
        static public void DoMerge(int[] numbers, int left, int mid, int right)
        {
            int[] temp = new int[25];
            int i, left_end, num_elements, tmp_pos;

            left_end = (mid - 1);
            tmp_pos = left;
            num_elements = (right - left + 1);

            while ((left <= left_end) && (mid <= right))
            {
                if (numbers[left] <= numbers[mid]) temp[tmp_pos++] = numbers[left++];
                else temp[tmp_pos++] = numbers[mid++];
            }

            while (left <= left_end) temp[tmp_pos++] = numbers[left++];

            while (mid <= right) temp[tmp_pos++] = numbers[mid++];

            for (i = 0; i < num_elements; i++)
            {
                numbers[right] = temp[right];
                right--;
            }
        }

        static public void MergeSort_Recursive(int[] numbers, int left, int right)
        {
            int mid;

            if (right > left)
            {
                mid = (right + left) / 2;
                MergeSort_Recursive(numbers, left, mid);
                MergeSort_Recursive(numbers, (mid + 1), right);

                DoMerge(numbers, left, (mid + 1), right);
            }
        }

        static void Main(string[] args)
        {
            Console.Write("Enter array length: ");
            int length = Int32.Parse(Console.ReadLine());

            int[] arr = new int[length];

            for (int i = 0; i < length; i++)
            {
                Console.Write("Enter {0} element: ", i);
                arr[i] = Int32.Parse(Console.ReadLine());
            }

            MergeSort_Recursive(arr, 0, arr.Length - 1);
```

```
            for (int i = 0; i < arr.Length; i++) Console.WriteLine(arr[i]);
        }
    }
}
```



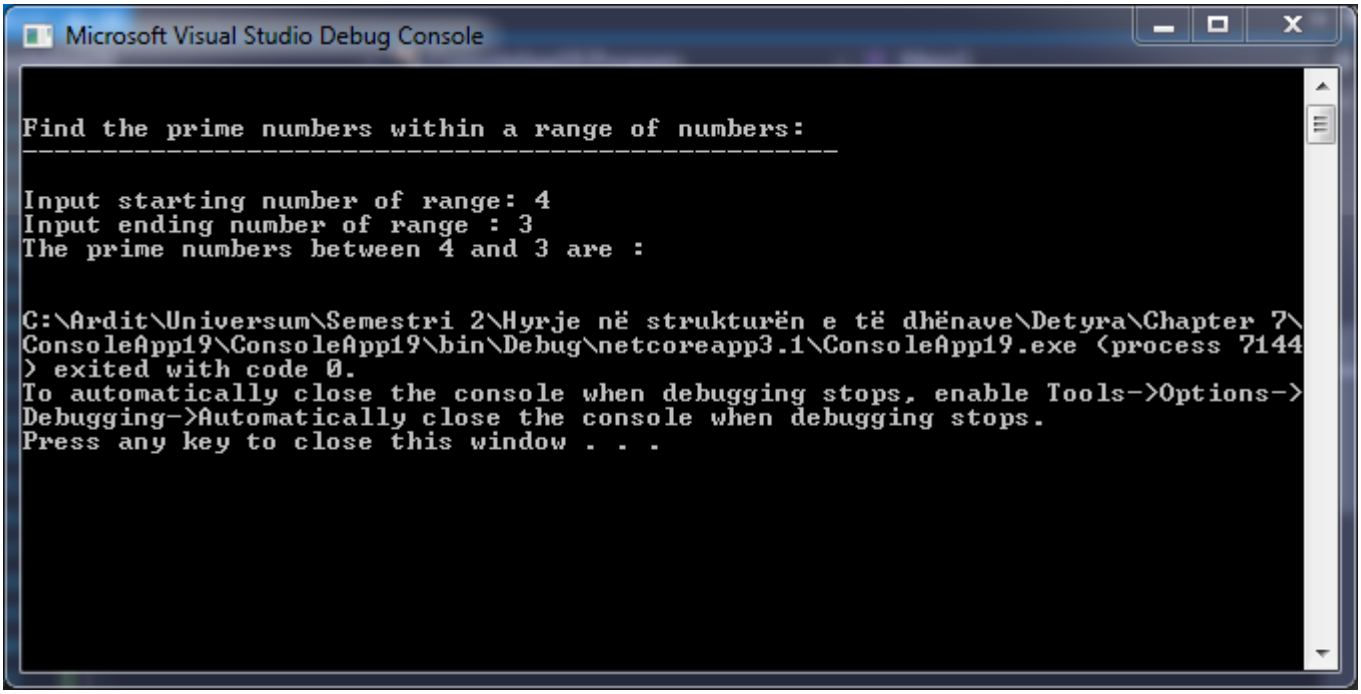
18. Write a program, which sorts an array of integer elements using a "**quick sort**" algorithm.

```
using System;

namespace ConsoleApp18
{
    class Program
    {
        public static void Quicksort(int[] elements, int left, int right)
        {
            int i = left, j = right;
            IComparable pivot = elements[(left + right) / 2];

            while (i <= j)
            {
                while (elements[i].CompareTo(pivot) < 0) i++;

                while (elements[j].CompareTo(pivot) > 0) j--;

                if (i <= j)
                {
                    int tmp = elements[i];
                    elements[i] = elements[j];
                    elements[j] = tmp;
                    i++;
                    j--;
                }
            }

            if (left < j) Quicksort(elements, left, j);

            if (i < right) Quicksort(elements, i, right);
        }

        static void Main(string[] args)
        {
            Console.Write("Enter array length: ");
            int length = Int32.Parse(Console.ReadLine());

            int[] arr = new int[length];

            for (int i = 0; i < length; i++)
            {
                Console.Write("Enter {0} element: ", i);
                arr[i] = Int32.Parse(Console.ReadLine());
            }

            Quicksort(arr, 0, arr.Length - 1);

            for (int i = 0; i < arr.Length; i++) Console.WriteLine(arr[i]);
        }
    }
}
```


Microsoft Visual Studio Debug Console
Enter array length: 3
Enter 0 element: 23
Enter 1 element: 12
Enter 2 element: 32
12
23
32
C:\Ardit\Universum\Semestri 2\Hyrje në strukturën e të dhënave\Detyra\Chapter 7\ConsoleApp18\ConsoleApp18\bin\Debug\netcoreapp3.1\ConsoleApp18.exe (process 7332) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

19. Write a program, which finds **all prime numbers** in the range [1…10,000,000].

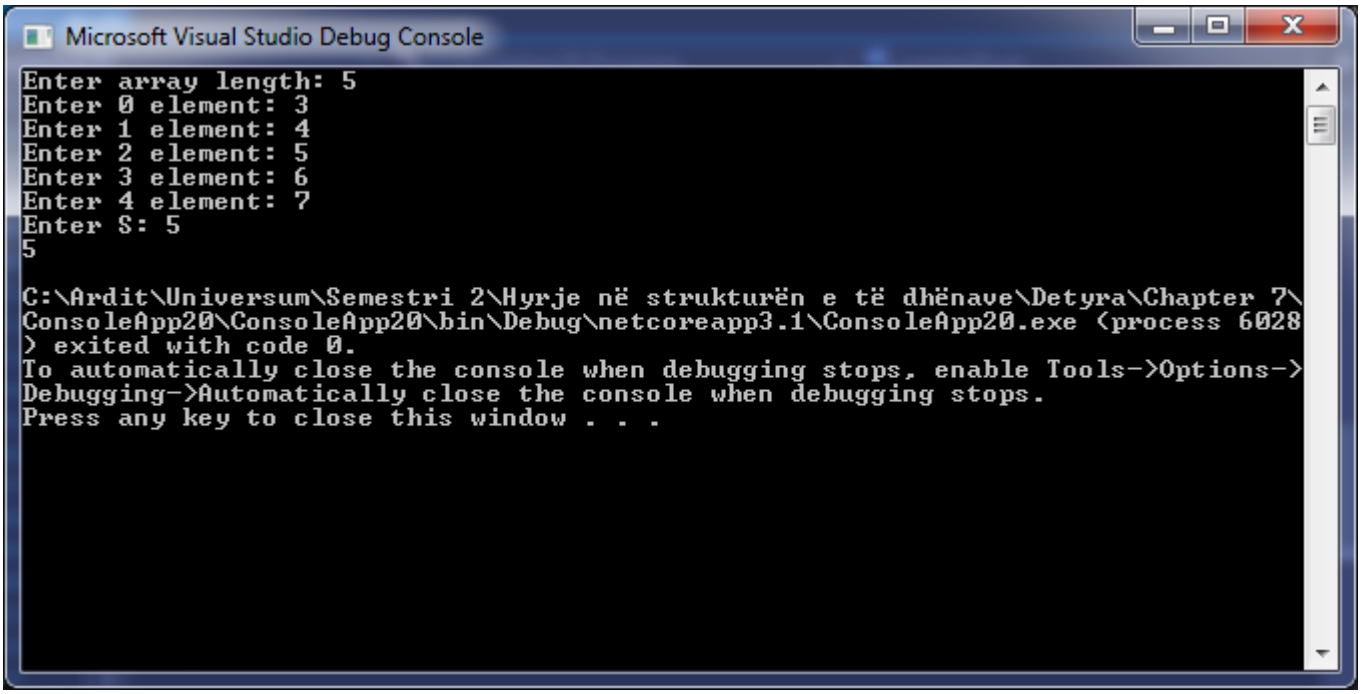
```
using System;

namespace ConsoleApp19
{
    class Program
    {
        public static void Main()
        {
            int num, i, ctr, stno, enno;

            Console.Write("\n\n");
            Console.Write("Find the prime numbers within a range of numbers:\n");
            Console.Write("-------------------------------------------------");
            Console.Write("\n\n");

            Console.Write("Input starting number of range: ");
            stno = Convert.ToInt32(Console.ReadLine());
            Console.Write("Input ending number of range : ");
            enno = Convert.ToInt32(Console.ReadLine());
            Console.Write("The prime numbers between {0} and {1} are : \n", stno, enno);

            for (num = stno; num <= enno; num++)
            {
                ctr = 0;

                for (i = 2; i <= num / 2; i++)
                {
                    if (num % i == 0)
                    {
                        ctr++;
                        break;
                    }
                }

                if (ctr == 0 && num != 1)
                    Console.Write("{0} ", num);
            }
            Console.Write("\n");
        }
    }

}
```


Microsoft Visual Studio Debug Console
Find the prime numbers within a range of numbers:
Input starting number of range: 4
Input ending number of range : 3
The prime numbers between 4 and 3 are :
C:\Ardit\Universum\Semestri 2\Hyrje në strukturën e të dhënave\Detyra\Chapter 7\ConsoleApp19\ConsoleApp19\bin\Debug\netcoreapp3.1\ConsoleApp19.exe (process 7144) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

20. Write a program, which checks whether there is a **subset** of given array of **N** elements, which has a **sum S**. The numbers `N, S` and the array values are read from the console. Same number can be used many times. Example: {2, 1, 2, 4, 3, 5, 2, 6}, S = 14 → yes (1 + 2 + 5 + 6 = 14)

```
using System;

namespace ConsoleApp20
{
    class Program
    {
        static int wantedSum;
        static bool solution = false;

        static void GenerateSubset(int[] arr, int[] subset, int index, int current, int
elementsInSubset)
        {
            if (index == elementsInSubset)
            {
                CheckSubsets(subset, elementsInSubset);
                return;
            }

            for (int i = current; i < arr.Length; i++)
            {
                subset[index] = arr[i];
                GenerateSubset(arr, subset, index + 1, i + 1, elementsInSubset);
            }
        }

        static void CheckSubsets(int[] subset, int elementsInSubset)
        {
            int sum = 0;

            for (int i = 0; i < elementsInSubset; i++) sum += subset[i];

            if (sum == wantedSum)
            {
                for (int i = 0; i < elementsInSubset; i++) Console.Write("{0} ", subset[i]);

                Console.WriteLine();
                solution = true;
            }
        }

        static void Main()
        {
            Console.Write("Enter array length: ");
            int length = int.Parse(Console.ReadLine());

            int[] arr = new int[length];

            for (int i = 0; i < length; i++)
            {
                Console.Write("Enter {0} element: ", i);
                arr[i] = int.Parse(Console.ReadLine());
            }

            Console.Write("Enter S: ");
            wantedSum = int.Parse(Console.ReadLine());

            int[] subset = new int[length];
```

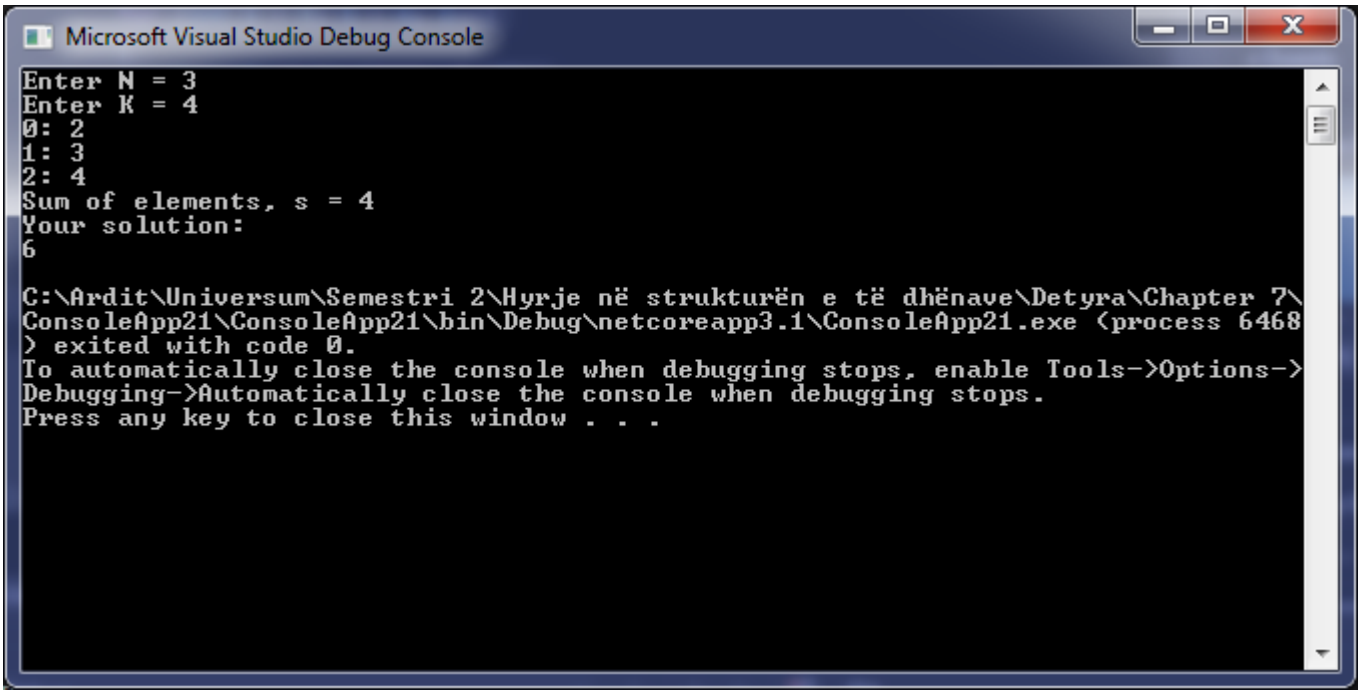
```
            for (int i = 1; i <= length; i++) GenerateSubset(arr, subset, 0, 0, i);

            if (!solution) Console.WriteLine("No subset with sum {0} found.", wantedSum);
        }
    }
}
```



21. Write a program which by given **N** numbers, **K** and **S**, finds **K** elements out of the **N** numbers, the sum of which is exactly **S** or says it is not possible.
Example: {3, **1**, 2, **4**, **9**, 6}, **S = 14**, **K = 3** → yes (1 + 2 + 4 = 14)

```
using System;

namespace ConsoleApp21
{
    class Program
    {
        public static int[] findSolution(int[] a, bool[] filter, int index, int s, int size)
        {
            if (index < a.Length)
            {
                filter[index] = true;
                int[] x = findSolution(a, filter, index + 1, s, size);

                if (x.Length > 0) return x;
                else
                {
                    filter[index] = false;
                    return findSolution(a, filter, index + 1, s, size);
                }
            }
            else
            {
                int sum = 0, count = 0;

                for (int i = 0; i < a.Length; i++)
                {
                    if (filter[i])
                    {
                        sum += a[i];
                        count++;
                    }
                }

                int[] solution = new int[0];

                if (sum == s && count == size)
                {
                    solution = new int[count];
                    count = 0;

                    for (int i = 0; i < a.Length; i++) if (filter[i]) solution[count++] = a[i];
                }
                return solution;
            }
        }

        static void Main(string[] args)
        {
            Console.Write("Enter N = ");
            int n = int.Parse(Console.ReadLine());

            Console.Write("Enter K = ");
            int size = int.Parse(Console.ReadLine());

            int[] arr = new int[n];

            for (int i = 0; i < n; i++)
            {
```

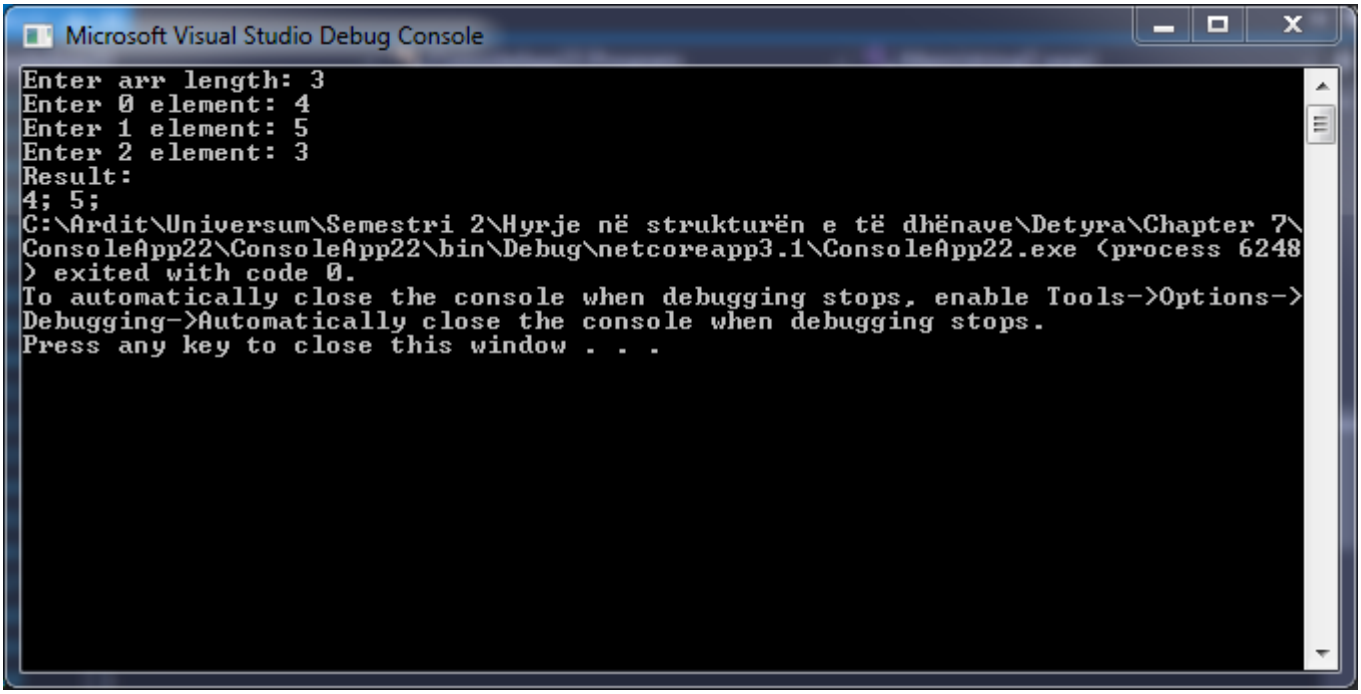
```
            Console.Write("{0}: ", i);
            arr[i] = int.Parse(Console.ReadLine());
        }

        Console.Write("Sum of elements, s = ");
        int s = int.Parse(Console.ReadLine());

        int[] solution = findSolution(arr, new bool[arr.Length], 0, s, size);

        Console.WriteLine("Your solution:");
        for (int i = 0; i < solution.Length; i++) Console.Write(solution[i] + "; ");
        Console.ReadLine();
    }
  }
}
```



22. Write a program, which reads an array of integer numbers from the console and **removes a minimal number of elements** in such a way that **the remaining array is sorted** in an increasing order.
Example: {6, **1**, 4, **3**, 0, **3**, 6, **4**, **5**} → {1, 3, 3, 4, 5}

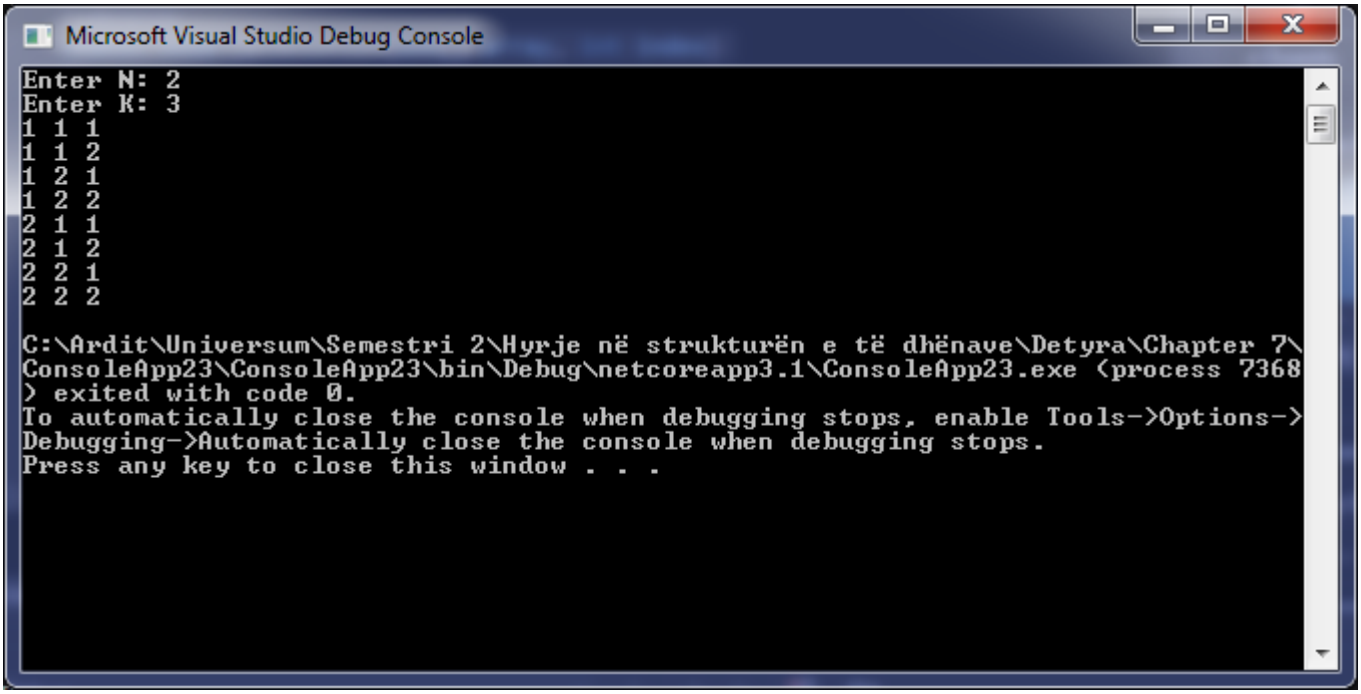
```
using System;

namespace ConsoleApp22
{
    class Program
    {
        static void Main(string[] args)
        {
            int subset = 0, longestLength = 0;

            Console.Write("Enter arr length: ");
            int length = Int32.Parse(Console.ReadLine());

            int[] arr = new int[length];

            for (int i = 0; i < length; i++)
            {
                Console.Write("Enter {0} element: ", i);
                arr[i] = Int32.Parse(Console.ReadLine());
            }

            int m = (1 << length);
            int[,] subsets = new int[m, length];

            for (int i = 0; i < m; i++)
                for (int j = 0; j < length; j++) subsets[i, j] = i / (m / 2 / (1 << j)) % 2;

            for (int i = 0; i < m; i++)
            {
                int max = -1000, count = 0;

                for (int j = 0; j < length; j++)
                {
                    if (subsets[i, j] > 0)
                    {
                        if (arr[j] >= max)
                        {
                            count++;
                            max = arr[j];
                        }
                        else
                        {
                            count = 0;
                            break;
                        }
                    }
                }

                if (longestLength < count)
                {
                    longestLength = count;
                    subset = i;
                }
            }

            Console.WriteLine("Result:");
            for (int i = 0; i < length; i++) if (subsets[subset, i] > 0) Console.Write(arr[i] + "; ");
        }
```

```
    }
}
```


Microsoft Visual Studio Debug Console
Enter arr length: 3
Enter 0 element: 4
Enter 1 element: 5
Enter 2 element: 3
Result:
4; 5;
C:\Ardit\Universum\Semestri 2\Hyrje në strukturën e të dhënave\Detyra\Chapter 7\ConsoleApp22\ConsoleApp22\bin\Debug\netcoreapp3.1\ConsoleApp22.exe (process 6248) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

23. Write a program, which reads the integer numbers `N` and `K` from the console and prints **all variations of K elements of the numbers in the interval [1…N]**.
Example: N = 3 → {1, 2, 3}, {1, 3, 2}, {2, 1, 3}, {2, 3, 1}, {3, 1, 2}, {3, 2, 1}

```
using System;

namespace ConsoleApp23
{
    class Program
    {
        public static int n;

        static void Main(string[] args)
        {
            Console.Write("Enter N: ");
            n = Int32.Parse(Console.ReadLine());

            Console.Write("Enter K: ");
            int k = Int32.Parse(Console.ReadLine());

            int[] arr = new int[k];

            recSolution(arr, 0);
        }

        static void recSolution(int[] array, int index)
        {
            if (index != array.Length)
                for (int i = 1; i <= n; i++)
                {
                    array[index] = i;
                    recSolution(array, index + 1);
                }
            else
            {
                for (int i = 0; i < array.Length; i++) Console.Write(array[i] + " ");
                Console.WriteLine();
            }
        }
    }
}
```


Microsoft Visual Studio Debug Console
Enter N: 2
Enter K: 3
1 1 1
1 1 2
1 2 1
1 2 2
2 1 1
2 1 2
2 2 1
2 2 2
C:\Ardit\Universum\Semestri 2\Hyrje në strukturën e të dhënave\Detyra\Chapter 7\ConsoleApp23\ConsoleApp23\bin\Debug\netcoreapp3.1\ConsoleApp23.exe (process 7368) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .