

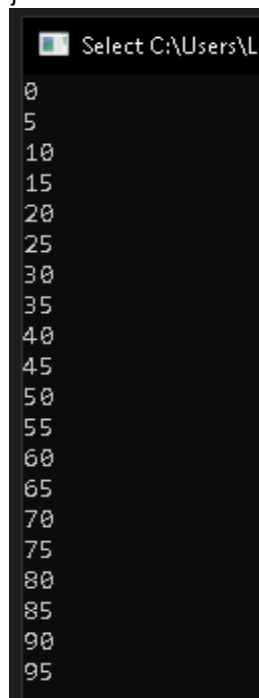
Chapter 7. Arrays

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 ex1
{
    class Program
    {
        static void Main(string[] args)
        {
            int[] array = new int[20];

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



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 ex2
{
    class Program
    {
        static void Main(string[] args)
        {
            bool equal = true;

            Console.Write("Enter length of 1st array : ");
            int length1 = Convert.ToInt32(Console.ReadLine());

            int[] firstArr = new int[length1];

            for(int i = 0; i<firstArr.Length; i++)
            {
                Console.Write("enter element : ");
                firstArr[i] = Convert.ToInt32(Console.ReadLine());
            }

            Console.Write("Enter length of 2nd array : ");
            int length2 = Convert.ToInt32(Console.ReadLine());

            int[] secondArr = new int[length2];

            for (int i = 0; i < secondArr.Length; i++)
            {
                Console.Write("enter element : ");
                secondArr[i] = Convert.ToInt32(Console.ReadLine());
            }

            if(firstArr.Length == secondArr.Length)
            {
                for(int i = 0; i < firstArr.Length; i++)
                {
                    if(firstArr[i] == secondArr[i])
                    {
                        equal = true;
                    }
                    else
                    {
                        equal = false;
                    }
                }
            }
        }
    }
}
```

```

    }
}
else
{
    equal = false;
}
if(equal == true) Console.WriteLine("Arrays are equal");
else Console.WriteLine("Arrays are not equal");
Console.ReadKey();
}
}
}

```

<pre> C:\Users\LumiDaK1NG\Desktop\UNI\pro Enter length of 1st array : 3 enter element : 1 enter element : 2 enter element : 5 Enter length of 2nd array : 3 enter element : 1 enter element : 2 enter element : 5 Arrays are equal </pre>	<pre> C:\Users\LumiDaK1NG\Desktop\UNI\pro Enter length of 1st array : 3 enter element : 1 enter element : 4 enter element : 7 Enter length of 2nd array : 3 enter element : 9 enter element : 6 enter element : 3 Arrays are not equal </pre>
---	---

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 ex3
```

```
{
```

```
    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 lexicographically first.");
```

```
            else if (arrA.Length < arrB.Length) Console.WriteLine("First array is lexicographically first.");
```

```
            else
```

```
            {
```

```
                for (int i = 0; i < arrA.Length; i++)
```

```
                {
```

```
                    if (arrA[i] < arrB[i])
```

```
                    {
```

```
                        Console.WriteLine("First array is lexicographically first.");
```

```
                        arrayEqual = false;
```

```
                        break;
```

```
                    }
```

```
                    if (arrA[i] > arrB[i])
```

```
                    {
```

```
                        Console.WriteLine("Second array is lexicographically first.");
```

```
                        arrayEqual = false;
```

```
                        break;
```

```
                    }
```

```
                }
```

```
            }
```

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

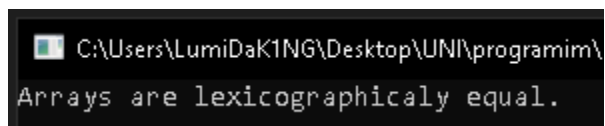
```
        }
```

```
        Console.ReadKey();
```

```
    }
```

```
}
```

```
}
```



```
C:\Users\LumiDaK1NG\Desktop\UNI\programim\  
Arrays are lexicographically equal.
```

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

```
using System;

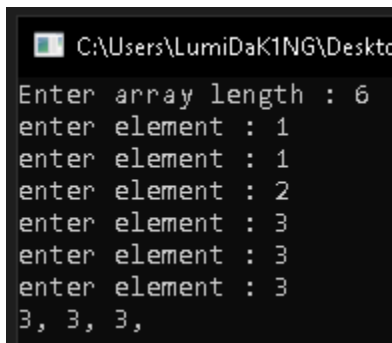
namespace ex4
{
    class Program
    {
        static void Main(string[] args)
        {
            int count = 1; int tempcount = 1, number = 0;

            Console.Write("Enter array length : ");
            int length = Convert.ToInt32(Console.ReadLine());

            int[] arr = new int[length];

            for(int i = 0; i < arr.Length; i++)
            {
                Console.Write("enter element : ");
                arr[i] = Convert.ToInt32(Console.ReadLine());
            }
            for(int i = 0; i < arr.Length-1; i++)
            {
                if (arr[i] == arr[i + 1]) tempcount++;
                else tempcount = 1;

                if (count < tempcount)
                {
                    count = tempcount;
                    number = arr[i];
                }
            }
            for(int i = 0; i < count; i++) Console.Write("{0}, ", number);
            Console.ReadKey();
        }
    }
}
```



```
C:\Users\LumiDaK1NG\Desktop
Enter array length : 6
enter element : 1
enter element : 1
enter element : 2
enter element : 3
enter element : 3
enter element : 3
3, 3, 3,
```

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

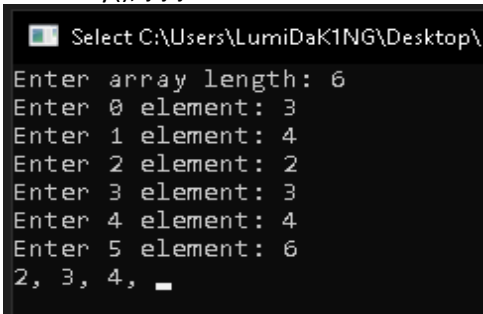
```
using System;
namespace ex5
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Enter array length: ");
            int length = Int32.Parse(Console.ReadLine());

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

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



Select C:\Users\LumiDaK1NG\Desktop\
Enter array length: 6
Enter 0 element: 3
Enter 1 element: 4
Enter 2 element: 2
Enter 3 element: 3
Enter 4 element: 4
Enter 5 element: 6
2, 3, 4, _

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. E.g.: {9, 6, 2, 7, 4, 7, 6, 5, 8, 4} à {2, 4, 6, 8}.

```
using System;

namespace ex6
{
    class Program
    {
        static void Main(string[] args)
        {
            int counter = 0, tempIndex, tempCounter;

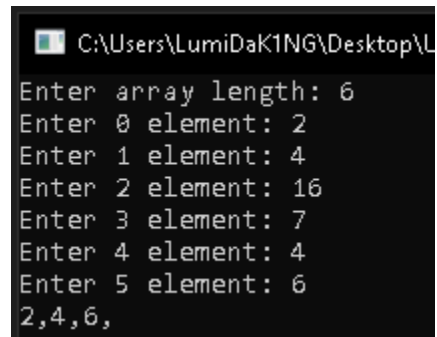
            Console.WriteLine("Enter array length: ");
            int length = Int32.Parse(Console.ReadLine());

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

            for (int i = 0; i < length; i++)
            {
                Console.WriteLine("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]);  
    Console.ReadKey();  
}  
}  
}
```



```
C:\Users\LumiDaK1NG\Desktop\L  
Enter array length: 6  
Enter 0 element: 2  
Enter 1 element: 4  
Enter 2 element: 16  
Enter 3 element: 7  
Enter 4 element: 4  
Enter 5 element: 6  
2,4,6,
```


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 ex7
{
    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());

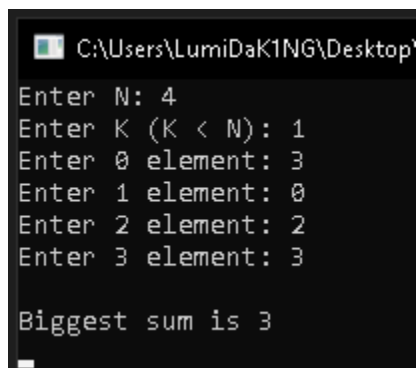
            int[] arr = new int[n];

            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);
            Console.ReadKey();
        }
    }
}
```



```
C:\Users\LumiDaK1NG\Desktop
Enter N: 4
Enter K (K < N): 1
Enter 0 element: 3
Enter 1 element: 0
Enter 2 element: 2
Enter 3 element: 3

Biggest sum is 3
```

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 ex8
{
    class Program
    {
        static void Main(string[] args)
        {
            int i, j, iMin, temp;

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

            int[] arr = new int[length];

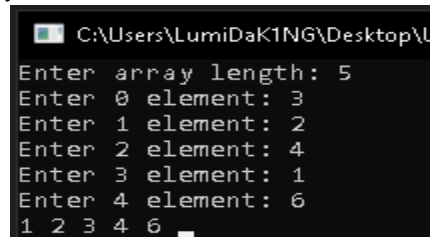
            for (i = 0; i < length; i++)
            {
                Console.WriteLine("Enter {0} element: ", i);
                arr[i] = Int32.Parse(Console.ReadLine());
            }

            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.WriteLine("{0} ", arr[i]);
            Console.ReadKey();
        }
    }
}
```



```
C:\Users\LumiDaK1NG\Desktop\L
Enter array length: 5
Enter 0 element: 3
Enter 1 element: 2
Enter 2 element: 4
Enter 3 element: 1
Enter 4 element: 6
1 2 3 4 6
```

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} à 11

```
using System;

namespace ex9
{
    class Program
    {
        static void Main(string[] args)
        {
            int sum = 0, tempSum;

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

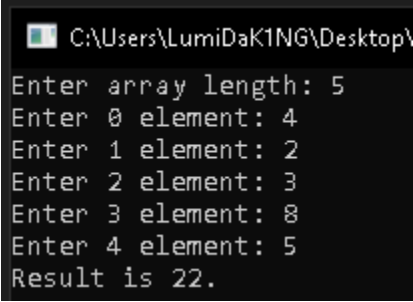
            int[] arr = new int[length];

            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);
            Console.ReadKey();
        }
    }
}
```



```
C:\Users\LumiDaK1NG\Desktop\
Enter array length: 5
Enter 0 element: 4
Enter 1 element: 2
Enter 2 element: 3
Enter 3 element: 8
Enter 4 element: 5
Result is 22.
```

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} à 4 (5 times).

```
using System;

namespace ex10
{
    class Program
    {
        static void Main(string[] args)
        {
            int counter = 0, tempCounter = 1, foundNumber = 0;

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

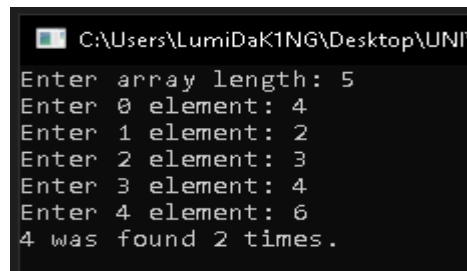
            int[] arr = new int[length];

            for (int i = 0; i < length; i++)
            {
                Console.WriteLine("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);
            Console.ReadKey(); } }
```



```
C:\Users\LumiDaK1NG\Desktop\UNI
Enter array length: 5
Enter 0 element: 4
Enter 1 element: 2
Enter 2 element: 3
Enter 3 element: 4
Enter 4 element: 6
4 was found 2 times.
```

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 à {4, 2, 5}.

```
using System;

namespace ex11
{
    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];

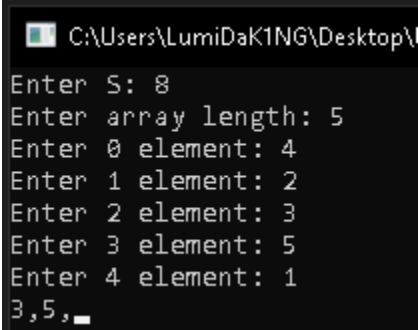
            for (int i = 0; i < length; i++)
            {
                Console.Write("Enter {0} element: ", i);
                arr[i] = Int32.Parse(Console.ReadLine());
            }

            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.");  
        Console.ReadKey();  
    }  
}  
}
```



```
C:\Users\LumiDaK1NG\Desktop\  
Enter S: 8  
Enter array length: 5  
Enter 0 element: 4  
Enter 1 element: 2  
Enter 2 element: 3  
Enter 3 element: 5  
Enter 4 element: 1  
3,5,
```

12. Write a program, which creates square matrices like those in the figures below and prints them formatted to the console. The size of the matrices will be read from the console. E.g. matrices with size of 4 x 4:

[clip_image025\[7\]](#)

```
using System;

namespace ex12
{
    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;

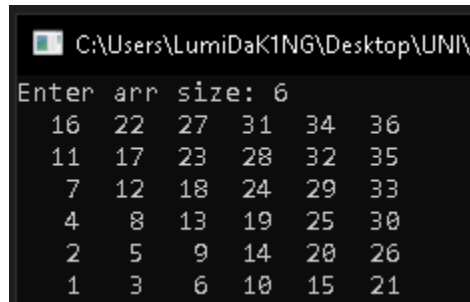
            for (int i = 1; i < counter - 1; i++)
            {
                for (int j = 1; j <= diagonalLength; j++)
                {
                    arr[posX, posY] = arr[prevX, prevY] - 1;
                    prevX = posX;
                    prevY = posY;
                    posX--;
               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

```



C:\Users\LumiDaK1NG\Desktop\UNI\

Enter arr size: 6

16	22	27	31	34	36
11	17	23	28	32	35
7	12	18	24	29	33
4	8	13	19	25	30
2	5	9	14	20	26
1	3	6	10	15	21

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 ex13
```

```
{
```

```
    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;
```

```
                    }
```

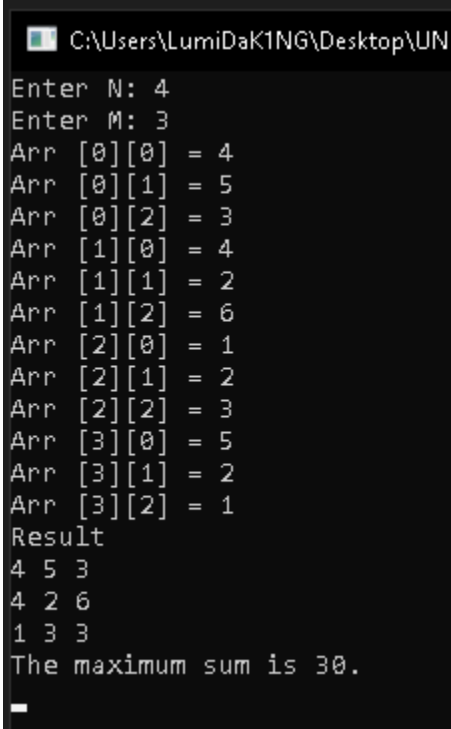
```
                }
```

```
            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);  
        Console.ReadKey();  
    }  
}  
}
```



```
C:\Users\LumiDaK1NG\Desktop\UN  
Enter N: 4  
Enter M: 3  
Arr [0][0] = 4  
Arr [0][1] = 5  
Arr [0][2] = 3  
Arr [1][0] = 4  
Arr [1][1] = 2  
Arr [1][2] = 6  
Arr [2][0] = 1  
Arr [2][1] = 2  
Arr [2][2] = 3  
Arr [3][0] = 5  
Arr [3][1] = 2  
Arr [3][2] = 1  
Result  
4 5 3  
4 2 6  
1 3 3  
The maximum sum is 30.  
_
```

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.

[clip_image027\[7\]](#)

```
using System;

namespace ex14
{
    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);
Console.ReadKey();
}
}
}

```

```

C:\Users\LumiDaK1NG\Desktop
Enter N: 5
Enter M: 4
Arr [0][0] = 3
Arr [0][1] = 1
Arr [0][2] = 4
Arr [0][3] = 7
Arr [1][0] = 5
Arr [1][1] = 6
Arr [1][2] = 3
Arr [1][3] = 2
Arr [2][0] = 5
Arr [2][1] = 4
Arr [2][2] = 3
Arr [2][3] = 4
Arr [3][0] = 5
Arr [3][1] = 21
Arr [3][2] = 4
Arr [3][3] = 8
Arr [4][0] = 3
Arr [4][1] = 5
Arr [4][2] = 4
Arr [4][3] = 2
5, 5, 5,

```

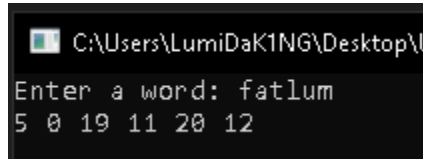
15. Write a program, which creates an array containing all Latin letters. The user inputs a word from the console and as result the program prints to the console the indices of the letters from the word.

```
using System;

namespace ex15
{
    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);
            Console.ReadKey();
        }
    }
}
```



A screenshot of a Windows command prompt window. The title bar shows the path 'C:\Users\LumiDaK1NG\Desktop\L'. The prompt displays the text 'Enter a word: fatlum' followed by the output '5 0 19 11 20 12' on the next line.

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

```
using System;

namespace ex16
{
    class Program
    {
        static void Main(string[] args)
        {

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

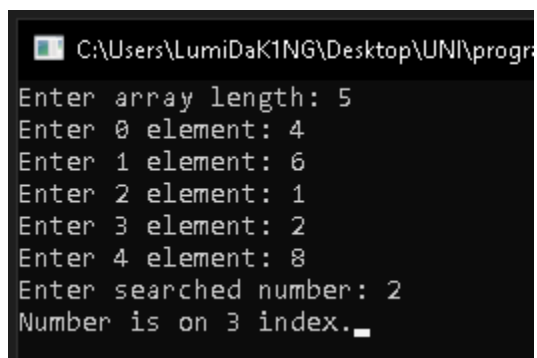
            int[] arr = new int[length];

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

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

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

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



A screenshot of a Windows command prompt window. The title bar shows the file path: C:\Users\LumiDaK1NG\Desktop\UNI\progr. The console output shows the program's execution: it prompts for an array length (5), then for five elements (4, 6, 1, 2, 8), then for a searched number (2), and finally outputs "Number is on 3 index." followed by a cursor.

```
C:\Users\LumiDaK1NG\Desktop\UNI\progr
Enter array length: 5
Enter 0 element: 4
Enter 1 element: 6
Enter 2 element: 1
Enter 3 element: 2
Enter 4 element: 8
Enter searched number: 2
Number is on 3 index._
```

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

```
using System;

namespace ex17
{
    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());

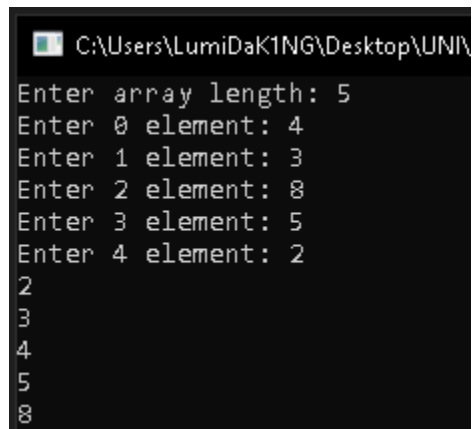
    int[] arr = new int[length];

    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]);
    Console.ReadKey();
}
}
}

```



```

C:\Users\LumiDaK1NG\Desktop\UNI\
Enter array length: 5
Enter 0 element: 4
Enter 1 element: 3
Enter 2 element: 8
Enter 3 element: 5
Enter 4 element: 2
2
3
4
5
8

```

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

```
using System;

namespace ex18
{
    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);
        }

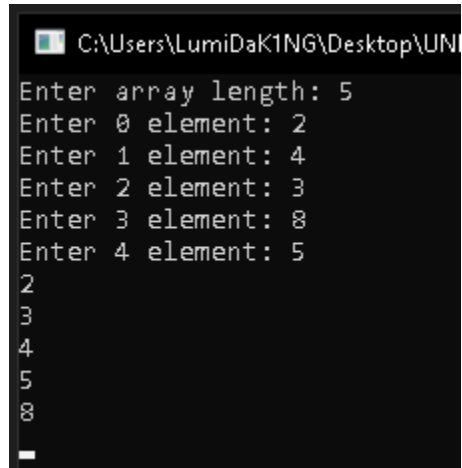
        static void Main(string[] args)
        {
            Console.Write("Enter array length: ");
            int length = Int32.Parse(Console.ReadLine());

            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]);
        Console.ReadKey();
    }
}
```



```
C:\Users\LumiDaK1NG\Desktop\UN...
Enter array length: 5
Enter 0 element: 2
Enter 1 element: 4
Enter 2 element: 3
Enter 3 element: 8
Enter 4 element: 5
2
3
4
5
8
_
```

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

```
using System;

namespace ex19
{
    class Program
    {
        static void Main(string[] args)
        {
            int numPrime = 0;
            bool isPrime = true;
            for (int i = 0; i <= 100; i++)
            {
                for (int j = 2; j <= 100; j++)
                {
                    if (i != j && i % j == 0)
                    {
                        isPrime = false;
                        break;
                    }
                }
                if (isPrime)
                {
                    Console.WriteLine(i);
                    numPrime += 1;
                }
                isPrime = true;
            }
            Console.WriteLine("Prime numbers 1 - 10,000,000 = {0}", numPrime);
            Console.ReadKey();
        }
    }
}
```

- Metoda e njejt eshte edhe deri ne 10,000,000, 1-100 jane gjithsej 26 numra Prime, kompjuterit i nevojitet kohe e gjate e procesimit per te shfaqe te gjithe numrat deri ne 10 milion.

C:\Users\LumiDaK1NG\Desktop\UNI\programim\semestr

```
1
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
53
59
61
67
71
73
79
83
89
97
Prime numbers 1 - 10,000,000 = 26
```

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 à yes (1 + 2 + 5 + 6 = 14)

```
using System;

namespace ex20
{
    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.WriteLine("Enter array length: ");
        int length = int.Parse(Console.ReadLine());

        int[] arr = new int[length];

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

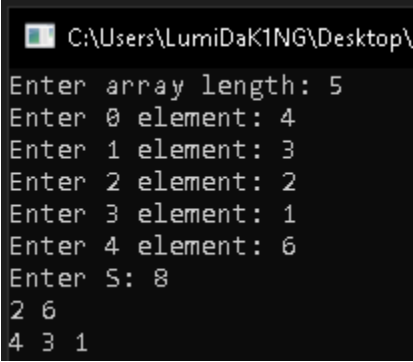
        Console.WriteLine("Enter S: ");
        wantedSum = int.Parse(Console.ReadLine());

        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);
        Console.ReadKey();
    }
}
}

```



```

C:\Users\LumiDaK1NG\Desktop\
Enter array length: 5
Enter 0 element: 4
Enter 1 element: 3
Enter 2 element: 2
Enter 3 element: 1
Enter 4 element: 6
Enter S: 8
2 6
4 3 1

```

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 à yes (1 + 2 + 4 = 14)

using System;

namespace ex21

{

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();
    Console.ReadKey();
}
}
}

```

```

C:\Users\LumiDak1NG\Desktop\UNI\p...
Enter N = 5
Enter K = 4
0: 2
1: 5
2: 8
3: 4
4: 5
Sum of elements, s = 10
Your solution:
8+2=10

```

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} à {1, 3, 3, 4, 5}

```
using System;

namespace ex22
{
    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
                    {

```

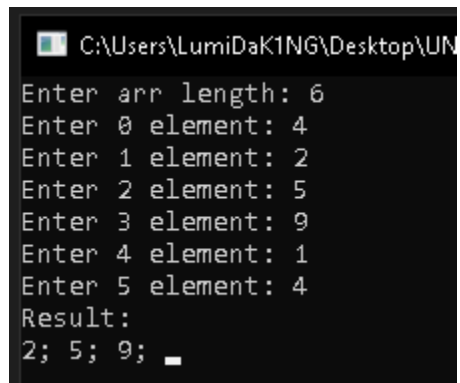
```

        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]
+ "; ");
Console.ReadKey();
}
}
}

```



```

C:\Users\LumiDaK1NG\Desktop\UN
Enter arr length: 6
Enter 0 element: 4
Enter 1 element: 2
Enter 2 element: 5
Enter 3 element: 9
Enter 4 element: 1
Enter 5 element: 4
Result:
2; 5; 9; 

```

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, K = 2 à {1, 1}, {1, 2}, {1, 3}, {2, 1}, {2, 2}, {2, 3}, {3, 1}, {3, 2}, {3, 3}.

```
using System;

namespace ex23
{
    class Program
    {
        public static int n;

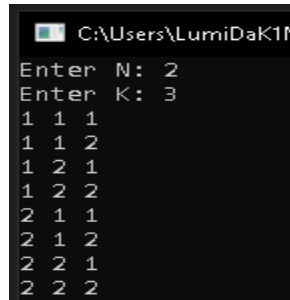
        static void Main(string[] args)
        {
            Console.Write("Enter N: ");
            n = Int32.Parse(Console.ReadLine());

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

            int[] arr = new int[k];

            recSolution(arr, 0);
            Console.ReadKey();
        }

        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();
            }
        }
    }
}
```



A screenshot of a Windows command prompt window. The title bar shows the path 'C:\Users\LumiDaK1N'. The console output is as follows:

```
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
```