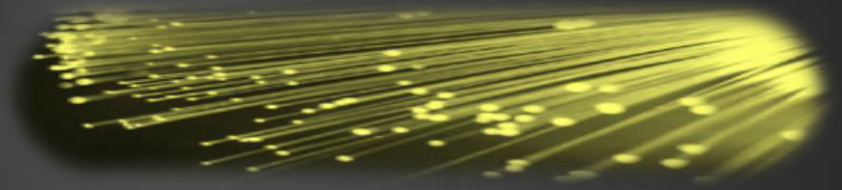
Telerik Academy

http://csharpfundamentals.telerik.com



Arrays

Processing Sequences of Elements

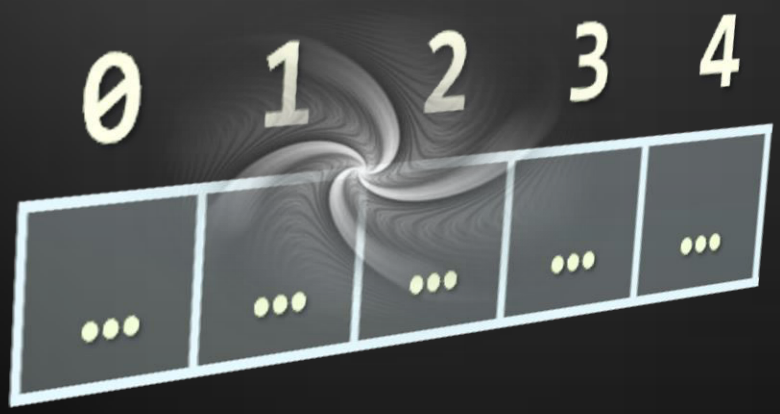
Telerik Software Academy
Learning & Development Team
http://academy.telerik.com

Telerik Academy

Table of Contents

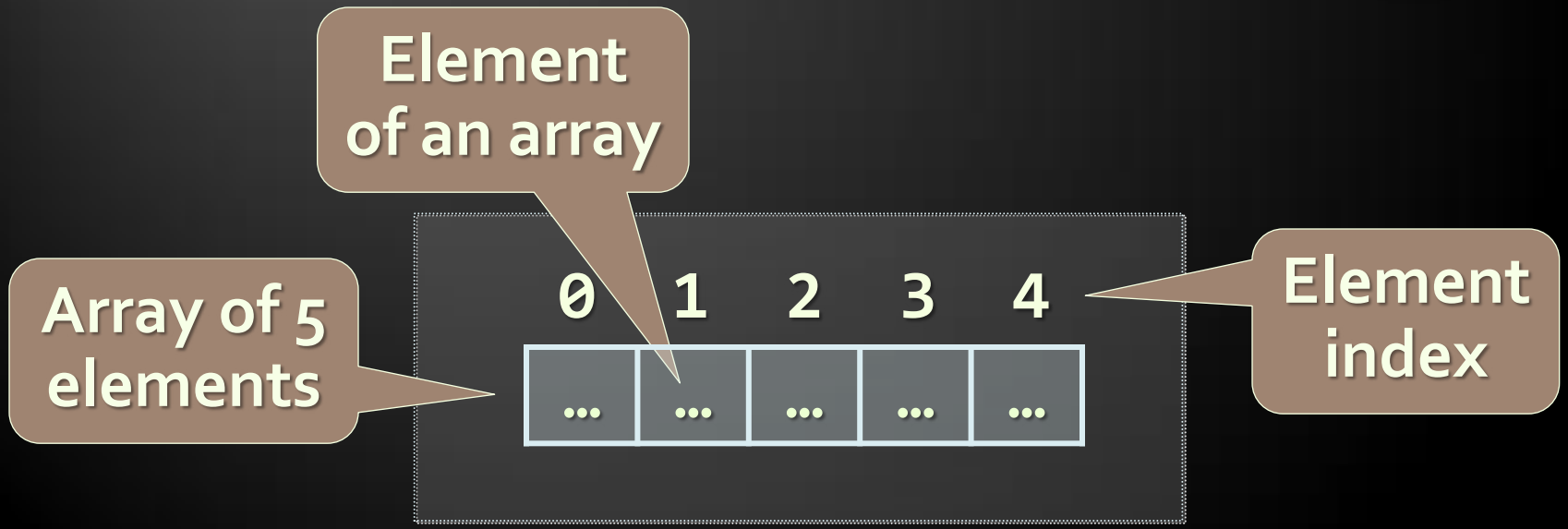
Telerik Academy

Declaring and Creating Arrays



What are Arrays?

- An array is a sequence of elements
  - All elements are of the same type
  - The order of the elements is fixed
  - Has fixed size (`Array.Length`)



Declaring Arrays

- Declaration defines the type of the elements
- Square brackets [ ] mean "array"
- Examples:
  - Declaring array of integers:

```
int[] myIntArray;
```

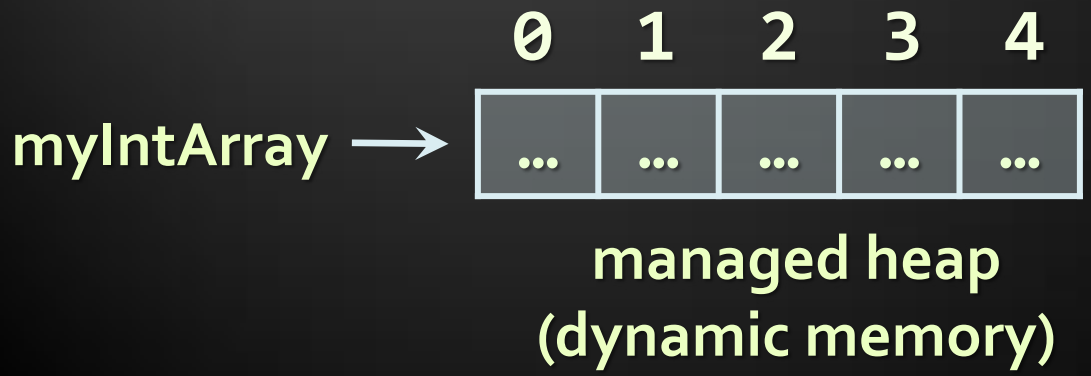
  - Declaring array of strings:

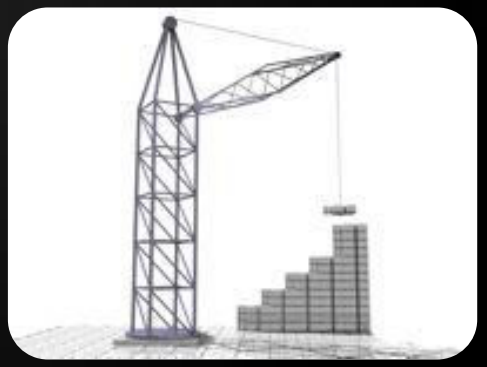
```
string[] myStringArray;
```

Creating Arrays

- Use the operator new
  - Specify array length
- Example creating (allocating) array of 5 integers:

```
myIntArray = new int[5];
```

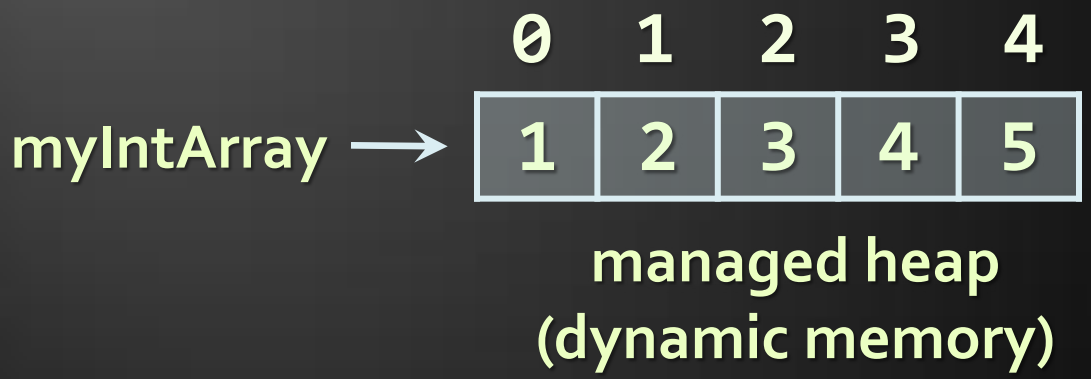




Creating and Initializing Arrays

- Creating and initializing can be done together:

```
myIntArray = {1, 2, 3, 4, 5};
```



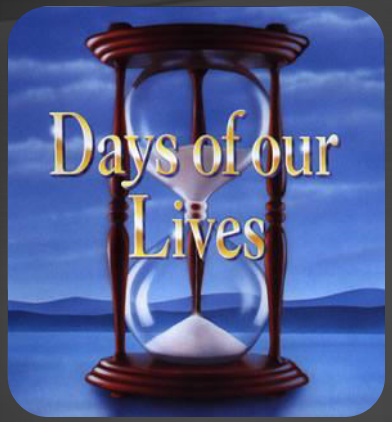
- The new operator is not required when using curly brackets initialization

Creating Array – Example

- Creating an array that contains the names of the days of the week

```
string[] daysOfWeek =
{
    "Monday",
    "Tuesday",
    "Wednesday",
    "Thursday",
    "Friday",
    "Saturday",
    "Sunday"
};
```

Telerik Academy



Days of Week

Live Demo

Accessing Array Elements

Read and Modify Elements by Index

Telerik Academy

How to Access Array Element?

- Array elements are accessed using the square brackets operator [ ] (indexer)
  - Array indexer takes element's index as parameter
  - The first element has index `0`
  - The last element has index `Length-1`
- Array elements can be retrieved and changed by the [ ] operator

Reversing an Array – Example

- Reversing the contents of an array

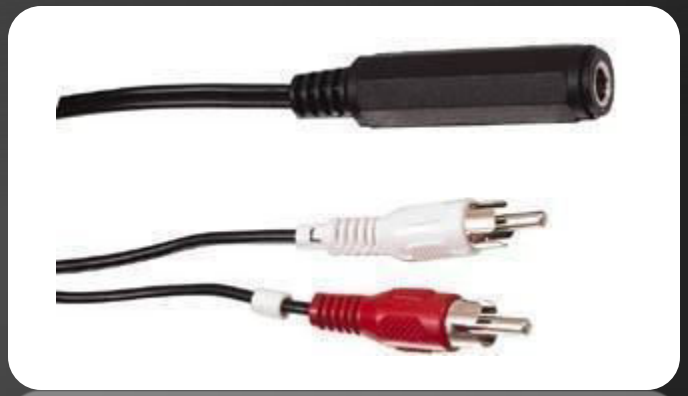
```
int[] array = new int[] {1, 2, 3, 4, 5};

// Get array size
int length = array.Length;

// Declare and create the reversed array
int[] reversed = new int[length];

// Initialize the reversed array
for (int index = 0; index < length; index++)
{
    reversed[length-index-1] = array[index];
}
```

Telerik Academy

Reversing an Array

Live Demo

Telerik Academy



Arrays: Input and Output

Reading and Printing Arrays on the Console

Telerik Academy

Reading Arrays From the Console

- First, read from the console the length of the array

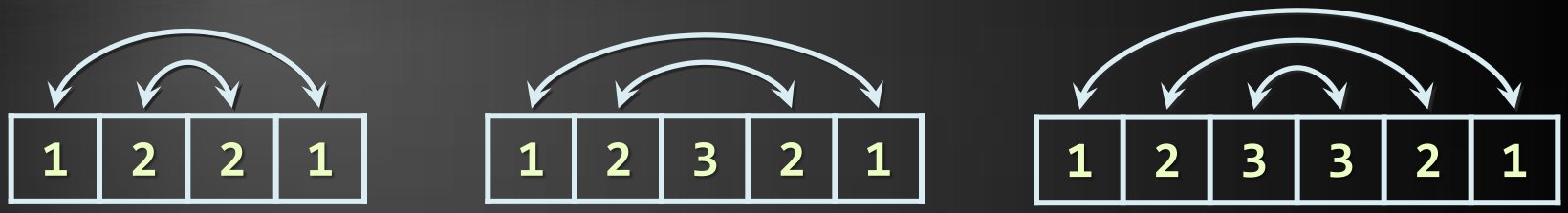
```
int n = int.Parse(Console.ReadLine());
```

- Next, create the array of given size and read its elements in a for loop

```
int[] arr = new int[n];
for (int i=0; i<n; i++)
{
    arr[i] = int.Parse(Console.ReadLine());
}
```

Symmetry Check – Example

- Read `int` array from the console and check if it is symmetric:




```
bool isSymmetric = true;
for (int i=0; i<array.Length/2; i++)
{
    if (array[i] != array[n-i-1])
    {
        isSymmetric = false;
    }
}
```

Telerik Academy

Symmetry Check

Live Demo

Telerik Academy

Printing Arrays on the Console

- Process all elements of the array
- Print each element to the console
- Separate elements with white space or a new line

```
string[] array = {"one", "two", "three"};

// Process all elements of the array
for (int index = 0; index < array.Length; index++)
{
    // Print each element on a separate line
    Console.WriteLine("element[{0}] = {1}",
        index, array[index]);
}
```

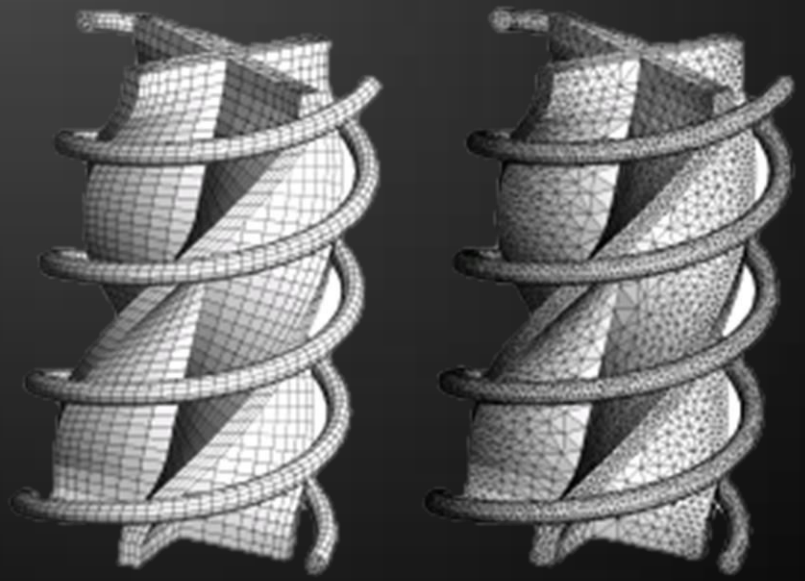
Telerik Academy

Printing Arrays

Live Demo

Telerik Academy

Processing Array Elements Using for and foreach



Processing Arrays: for Statement

- Use for loop to process an array when
  - Need to keep track of the index
  - Processing is not strictly sequential from the first to the last element
- In the loop body use the element at the loop index (array[index]):

```
for (int index = 0; index < array.Length; index++)
{
   squares[index] = array[index] * array[index];
}
```

Processing Arrays Using for Loop – Examples

- Printing array of integers in reversed order:

```
Console.WriteLine("Reversed: ");
for (int i = array.Length-1; i >= 0; i--)
{
    Console.Write(array[i] + " ");
}
// Result: 5 4 3 2 1
```

- Initialize all array elements with their corresponding index number:

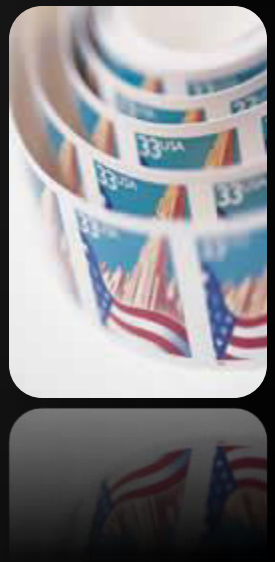
```
for (int index = 0; index < array.Length; index++)
{
    array[index] = index;
}
```

Processing Arrays: foreach

- How foreach loop works?

```
foreach (type value in array)
```

  - type – the type of the element
  - value – local name of variable
  - array – processing array



- Used when no indexing is needed
  - All elements are accessed one by one
  - Elements can not be modified (read only)

Processing Arrays Using foreach – Example

- Print all elements of a `string[]` array:

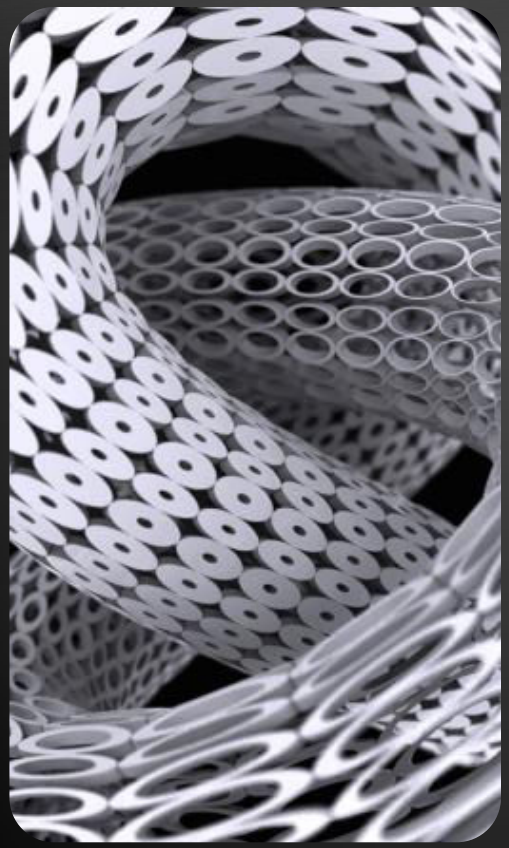
```
string[] capitals =
{
    "Sofia",
    "Washington",
    "London",
    "Paris"
};
foreach (string capital in capitals)
{
    Console.WriteLine(capital);
}
```

Telerik Academy

Processing Arrays

Live Demo

Telerik Academy



Resizable Arrays

List<T>

Lists (Resizable Arrays)

- `List<T>` – array that can resize dynamically
  - When adding or removing elements
  - Also have indexers `[]` (like arrays)
  - `T` is the type that the list will hold
    - E.g. `List<int>` will hold integers
    - `List<object>` will hold objects
- Basic methods and properties
  - `Add(T element)` – adds new element to the end
  - `Remove(element)` – removes the element
  - `Count` – returns the current size of the list

List Example

```
List<int> intList = new List<int>();
for( int i=0; i<5; i++)
{
   intList.Add(i);
}
```

- Is the same as:

```
int[] intArray = new int[5];
for( int i=0; i<5; i++)
{
   intArray[i] = i;
}
```

- The main difference
  - When using lists we don't have to know the exact number of elements

Lists vs. Arrays

- Lets have an array with capacity of 5 elements

```
int[] intArray = new int[5];
```

- If we want to add a sixth element (we have already added 5) we have to manually resize

```
int[] copyArray = intArray;
int[] intArray = new int[6];
for (int i = 0; i < 5; i++)
{
   intArray[i] = copyArray[i];
}
intArray[5] = newValue;
```

- With `List<T>` we simply call

```
list.Add(newValue);
```

Telerik Academy

Lists <T>

Live Demo

How The List<T> Works?

- Why adding new elements is not slow?
  - When adding n elements in `List<T>` it resizes itself $\log_{(2)}n$ times instead of n
- Initially a new List<T> has size of `0` elements
  - Counter for total capacity (Capacity)
  - Counter for number of used capacity (Count)
  - When created, both properties of the list have values of `0`
  - When adding the first element `Count` becomes `1` and `Capacity` becomes `4`

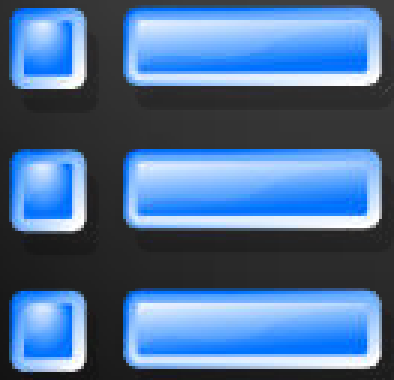
How The List<T> Works? (2)

- Initially the `List<T>` is empty
  - When adding new element it is resized
  - But not every time
    - Only when it is needed
- Lets have a list with 3 elements
  - It looks like this:
  - When we add new element it is appended to the end
  - Adding a fifth element doubles the Capacity of the list

Telerik Academy

Resizing Lists

Live Demo



Telerik Academy

Copying Arrays

The Array Class

Copying Arrays

- Sometimes we must copy the values from one array to another one
  - If we do it the intuitive way we would copy not only the values but the reference to the array
    - Changing some of the values in one array will affect the other

```
int[] copyArray = array;
```

  - The way to avoid this is using `Clone()`

```
int[] copyArray = (int[])array.Clone();
```

    - This way only the values will be copied but not the reference

Summary

- Arrays are a fixed-length sequences of elements of the same type
- Array elements are accessible by index
  - Can be read and modified
- Iteration over array elements can be done with `for` and `foreach` loops
- `List<T>` holds resizable arrays
  - Good when we don't know the number of elements initially

Telerik Academy

Arrays

Questions?

http://csharpfundamentals.telerik.com

Exercises

1. Write a program that allocates array of 20 integers and initializes each element by its index multiplied by 5. Print the obtained array on the console.
2. Write a program that reads two arrays from the console and compares them element by element.
3. Write a program that compares two char arrays lexicographically (letter by letter).
4. Write a program that finds the maximal sequence of equal elements in an array.

   Example: {2, 1, 1, 2, 3, 3, 2, 2, 2, 1} → {2, 2, 2}.

Exercises (2)

5. Write a program that finds the maximal increasing sequence in an array. Example:
   {3, 2, 3, 4, 2, 2, 4} → {2, 3, 4}.
6. Write a program that reads two integer numbers N and K and an array of N elements from the console. Find in the array those K elements that have maximal sum.
7. Sorting an array means to arrange its elements in increasing order. Write a program to sort an array. Use the "selection sort" algorithm: Find the smallest element, move it at the first position, find the smallest from the rest, move it at the second position, etc.

Exercises (3)

8. Write a program that finds the sequence of maximal sum in given array. Example:

   {2, 3, -6, -1, 2, -1, 6, 4, -8, 8} → {2, -1, 6, 4}

   Can you do it with only one loop (with single scan through the elements of the array)?

9. Write a program that finds the most frequent number in an array. Example:

   {4, 1, 1, 4, 2, 3, 4, 4, 1, 2, 4, 9, 3} → 4 (5 times)

10. Write a program that finds in given array of integers a sequence of given sum S (if present). Example:
    {4, 3, 1, 4, 2, 5, 8}, S=11 → {4, 2, 5}

Exercises (6)

11. Write a program that finds the index of given element in a sorted array of integers by using the binary search algorithm (find it in Wikipedia).
12. Write a program that creates an array containing all letters from the alphabet (A-Z). Read a word from the console and print the index of each of its letters in the array.
13. * Write a program that sorts an array of integers using the merge sort algorithm (find it in Wikipedia).
14. Write a program that sorts an array of strings using the quick sort algorithm (find it in Wikipedia).

Exercises (7)

15. Write a program that finds all prime numbers in the range [1…10 000 000]. Use the sieve of Eratosthenes algorithm (find it in Wikipedia).
16. * We are given an array of integers and a number S. Write a program to find if there exists a subset of the elements of the array that has a sum S.
    Example:

    arr={2, 1, 2, 4, 3, 5, 2, 6}, S=14 → yes (1+2+5+6)

17. * Write a program that reads three integer numbers N, K and S and an array of N elements from the console. Find in the array a subset of K elements that have sum S or indicate about its absence.

Exercises (8)

18. * Write a program that reads an array of integers and removes from it a minimal number of elements in such way that the remaining array is sorted in increasing order. Print the remaining sorted array. Example:

    {6, 1, 4, 3, 0, 3, 6, 4, 5} → {1, 3, 3, 4, 5}

19. * Write a program that reads a number N and generates and prints all the permutations of the numbers [1 … N]. Example:

    n = 3 → {1, 2, 3}, {1, 3, 2}, {2, 1, 3}, {2, 3, 1}, {3, 1, 2}, {3, 2, 1}

Exercises (9)

20. Write a program that reads two numbers N and K and generates all the variations of K elements from the set [1..N]. Example:

    N = 3, K = 2 → {1, 1}, {1, 2}, {1, 3}, {2, 1}, {2, 2}, {2, 3}, {3, 1}, {3, 2}, {3, 3}

21. Write a program that reads two numbers N and K and generates all the combinations of K distinct elements from the set [1..N]. Example:

    N = 5, K = 2 → {1, 2}, {1, 3}, {1, 4}, {1, 5}, {2, 3}, {2, 4}, {2, 5}, {3, 4}, {3, 5}, {4, 5}

Free Trainings @ Telerik Academy

- "C# Programming @ Telerik Academy
  - csharpfundamentals.telerik.com

- Telerik Software Academy
  - academy.telerik.com

- Telerik Academy @ Facebook
  - facebook.com/TelerikAcademy

- Telerik Software Academy Forums
  - forums.academy.telerik.com