Arrays

Processing Sequences of Elements

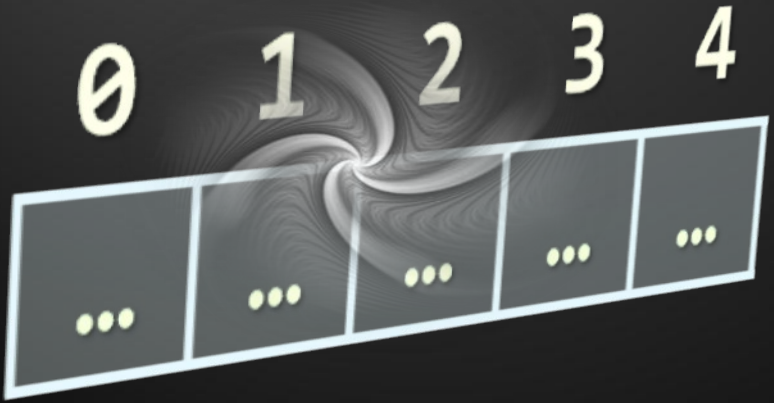
Svetlin Nakov

Telerik Corporation

www.telerik.com

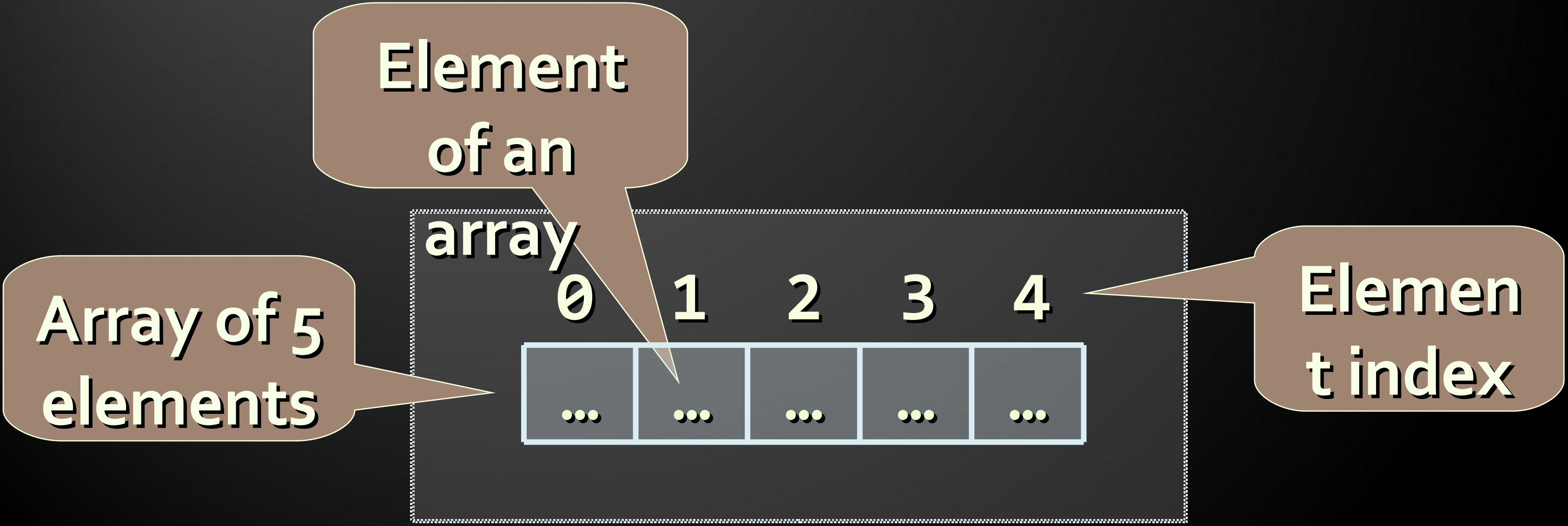
Table of Contents

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;
```

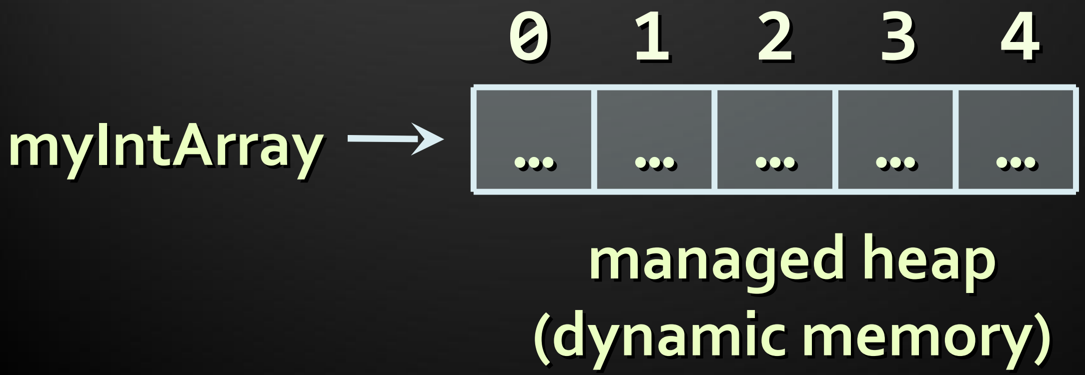
  - Declaring array of strings:

```
string[] myStringArray;
```

Creating Arrays

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

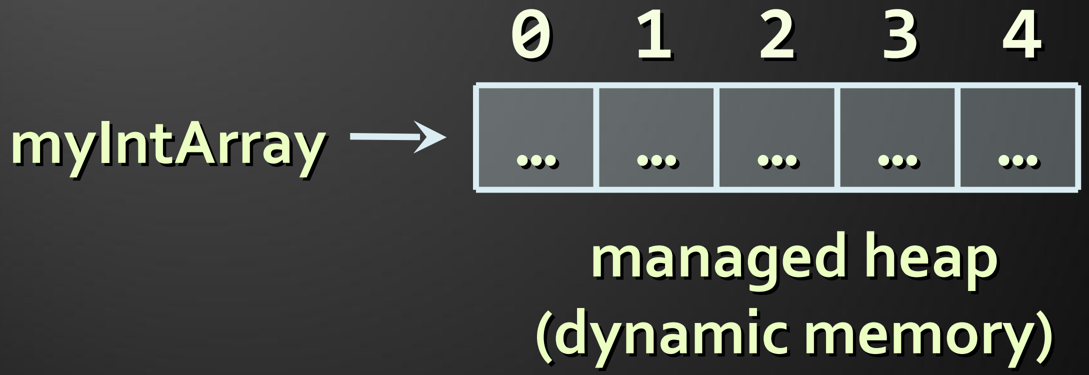
```
myIntArray = new int[5];
```



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

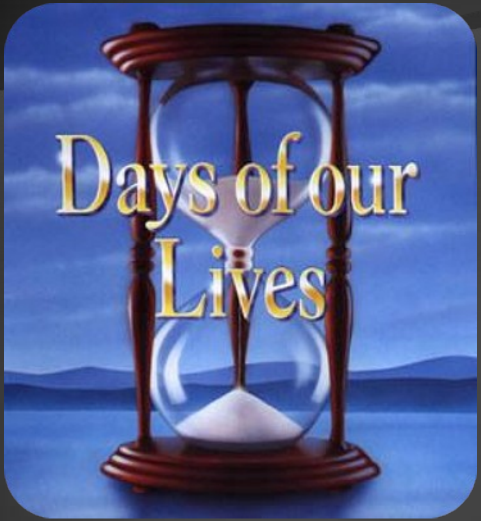
Creating Array – Example

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

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

Days of Week

Live Demo



Accessing Array Elements

Read and Modify Elements by Index

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

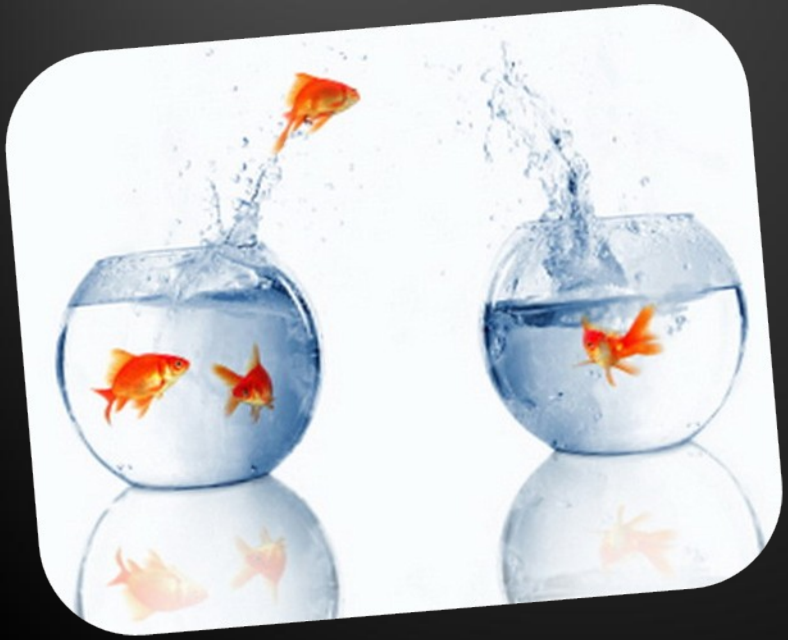
Reversing an Array – Example

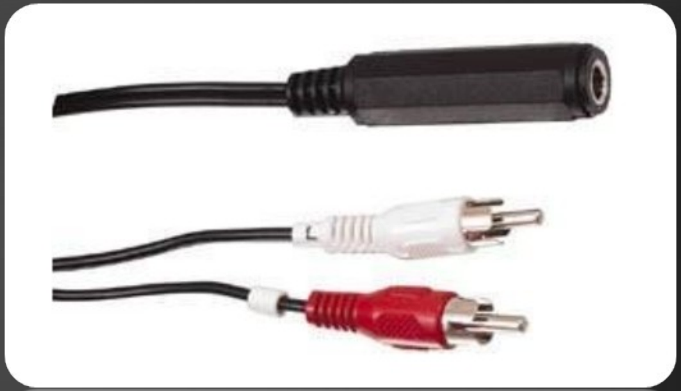
- Reversing the contents of an array

```
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];
}
```

Reversing an Array

Live Demo





Arrays: Input and Output

Reading and Printing Arrays on the Console

Reading Arrays From the Console

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

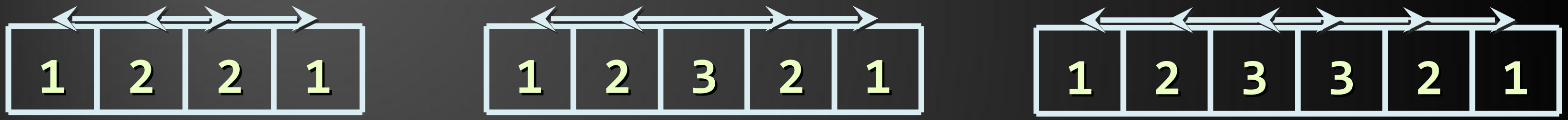
```
int n = int.Parse(Console.ReadLine());
```

- 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+1)/2; i++)
{
    if (array[i] != array[n-i-1])
    {
        isSymmetric = false;
    }
}
```

Symmetry Check

Live Demo

telerik

Printing Arrays on the Console

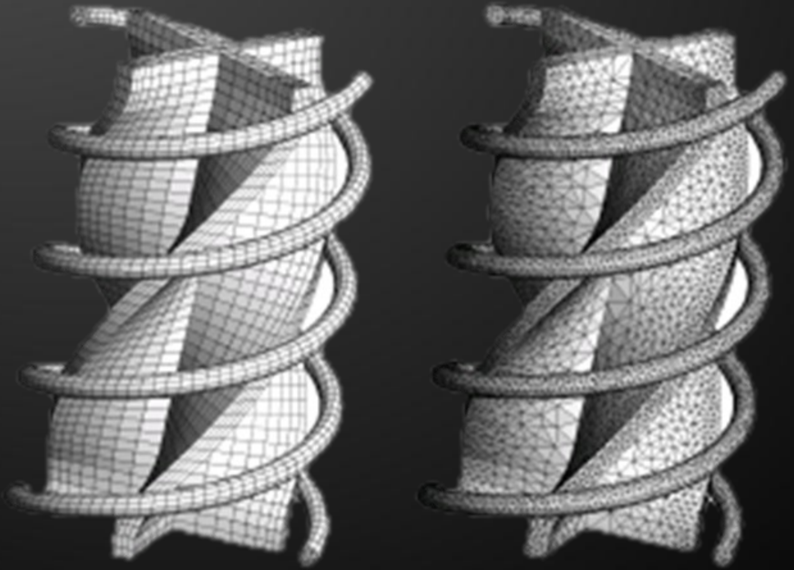
- Process all elements of the array
- Print each element to the console
- Separate elements with white space or a new line

```
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]);
}
```

Printing Arrays

Live Demo

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];
}
```

telerik

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-1; 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:

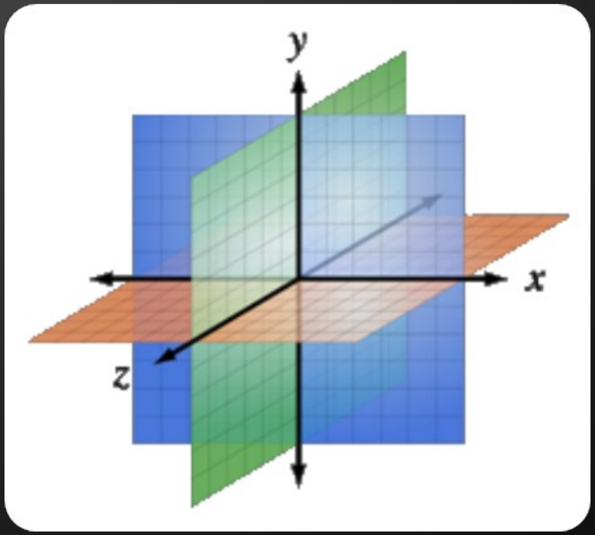
```
string[] capitals =
{
    "Sofia",
    "Washington",
    "London",
    "Paris"
};
foreach (string capital in capitals)
{
    Console.WriteLine(capital);
}
```

Processing Arrays

Live Demo

Multidimensional Arrays

Using Array of Arrays, Matrices and Cubes



telerik

What is Multidimensional Array?

- Multidimensional arrays have more than one dimension (2, 3, …)
  - The most important multidimensional arrays are the 2-dimensional
    - Known as matrices or tables
- Example of matrix of integers with 2 rows and 4 columns:

	0	1	2	3
0	5	0	-2	4
1	5	6	7	8

Declaring and Creating Multidimensional Arrays

- Declaring multidimensional arrays:

```
int[,] intMatrix;
float[,] floatMatrix;
string[,,] strCube;
```

- Creating a multidimensional array
  - Use new keyword
  - Must specify the size of each dimension

```
int[,] intMatrix = new int[3, 4];
float[,] floatMatrix = new float[8, 2];
string[,,] stringCube = new string[5, 5, 5];
```

telerik

Initializing Multidimensional Arrays with Values

- Creating and initializing with values multidimensional array:

```
int[,] matrix =
{
    {1, 2, 3, 4}, // row 0 values
    {5, 6, 7, 8}, // row 1 values
}; // The matrix size is 2 x 4 (2 rows, 4 cols)
```

  - Matrices are represented by a list of rows
    - Rows consist of list of values
  - The first dimension comes first, the second comes next (inside the first)

Accessing The Elements of Multidimensional Arrays

- Accessing N-dimensional array element:

```
nDimensionalArray[index1, … , indexn]
```

- Getting element value example:

```
int[,] array = {{1, 2}, {3, 4}}
int element11 = array[1, 1]; //element11 = 4
```

- Setting element value example:

Number of rows

```
int[,] array = new int[3, 4];
for (int row=0; row<array.GetLength(0); row++)
  for (int col=0; col<array.GetLength(1); col++)
    array[row, col] = row + col;
```

Number of

Reading Matrix – Example

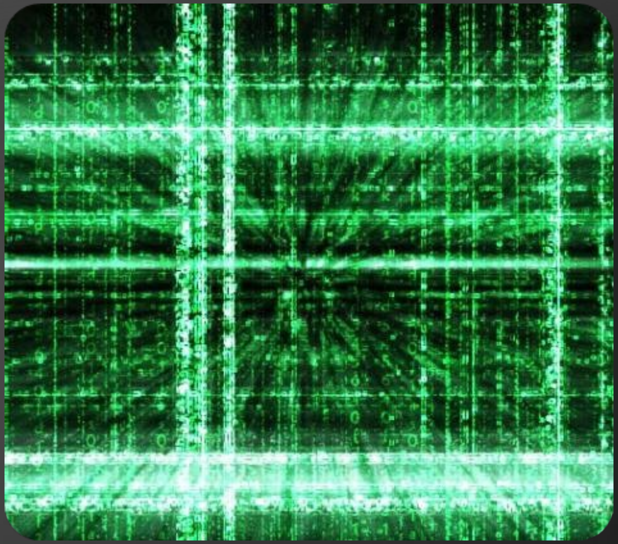
- Reading a matrix from the console

```
int rows = int.Parse(Console.ReadLine());
int columns = int.Parse(Console.ReadLine());
int[,] matrix = new int[rows, columns];
String inputNumber;
for (int row=0; row<rows; row++)
{
  for (int column=0; column<cols; column++)
  {
    Console.Write("matrix[{0},{1}] = ", row, column);
    inputNumber = Console.ReadLine();
    matrix[row, column] = int.Parse(inputNumber);
  }
}
```

Printing Matrix – Example

- Printing a matrix on the console:

```
for (int row=0; row<matrix.GetLength(0); row++)
{
  for (int col=0; col<matrix.GetLength(1); col++)
  {
    Console.Write("{0} ", matrix[row, col]);
  }
  Console.WriteLine();
}
```



Reading and Printing Matrices

Live Demo

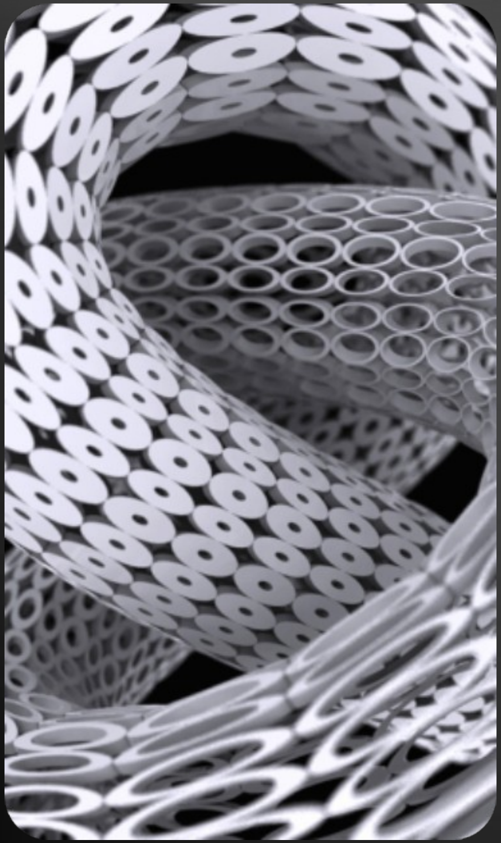
Maximal Platform – Example

- Finding a 2 x 2 platform in a matrix with a maximal sum of its elements

```
int[,] matrix = {
  {7, 1, 3, 3, 2, 1},
  {1, 3, 9, 8, 5, 6},
  {4, 6, 7, 9, 1, 0}
};
int bestSum = int.MinValue;
for (int row=0; row<matrix.GetLength(0)-1; row++)
  for (int col=0; col<matrix.GetLength(1)-1; col++)
  {
    int sum = matrix[row, col] + matrix[row, col+1]
      + matrix[row+1, col] + matrix[row+1, col+1];
    if (sum > bestSum)
      bestSum = sum;
  }
```

Maximal Platform

Live Demo



Dynamic Arrays

List<T>

Lists

- `Lists` are arrays that resize dynamically
  - When adding or removing elements
  - Also have indexers (like `Array`)
  - 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 do

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

- With List we simply do

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

Lists <T>

Live Demo

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 `Array.Copy()`

```
Array.Copy(sourceArray, copyArray);
```

    - 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
- Matrices (2-dimensional arrays) are very useful for presenting tabular data

Questions?

telerik

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)

1. Write a program that finds the maximal increasing sequence in an array. Example:
   {3, 2, 3, 4, 2, 2, 4} → {2, 3, 4}.
2. 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.
3. 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)

1. 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)?

4. 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)

a 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}

telerik

Exercises (4)

1. Write a program that fills and prints a matrix of size (n, n) as shown below: (examples for n = 4)

a)

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

b)

1	8	9	16
2	7	10	15
3	6	11	14
4	5	12	13

c)

7	11	14	16
4	8	12	15
2	5	9	13
1	3	6	10

d)

1	12	11	10
2	13	16	9
3	14	15	8
4	5	6	7

Exercises (5)

- Write a program that reads a rectangular matrix of size N x M and finds in it the square 3 x 3 that has maximal sum of its elements.
- We are given a matrix of strings of size N x M. Sequences in the matrix we define as sets of several neighbor elements located on the same line, column or diagonal. Write a program that finds the longest sequence of equal strings in the matrix. Examples:

ha	fifi	ho	hi
fo	ha	hi	xx
xxx	ho	ha	xx

$\longrightarrow$ ha, ha, ha

s	qq	s
pp	pp	s
pp	qq	s

$\longrightarrow$ s, s, s

Exercises (6)

1. 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).
2. 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.
3. Write a program that sorts an array of integers using the merge sort algorithm (find it in Wikipedia).
4. Write a program that sorts an array of strings using the quick sort algorithm (find it in Wikipedia).

Exercises (7)

z 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).

i * 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)

4. * 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)

1. * 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, $\boxed{1}$, 4, $\boxed{3}$, 0, $\boxed{3}$, 6, $\boxed{4}$, $\boxed{5}$} → {1, 3, 3, 4, 5}

3. * 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)

1. 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}

3. 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}

telerik

Exercises (10)

1. Write a program that fills a matrix of size (N, N) as shown in the examples (for N=4):

a)

16	15	13	10
14	12	9	6
11	8	5	3
7	4	2	1

b)

7	11	14	16
4	8	12	15
2	5	9	13
1	3	6	10

*c)

1	12	11	10
2	13	16	9
3	14	15	8
4	5	6	7

*d)

10	11	12	13
9	2	3	14
8	1	4	15
7	6	5	16

Exercises (11)

1. * Write a program that finds the largest area of equal neighbor elements in a rectangular matrix and prints its size. Example:

1	3	2	2	2	4
3	3	3	2	4	4
4	3	1	2	3	3
4	3	1	3	3	1
4	3	3	3	1	1

$\longrightarrow$ 13

Hint: you can use the algorithm "Depth-first search" or "Breadth-first search" (find them in Wikipedia).