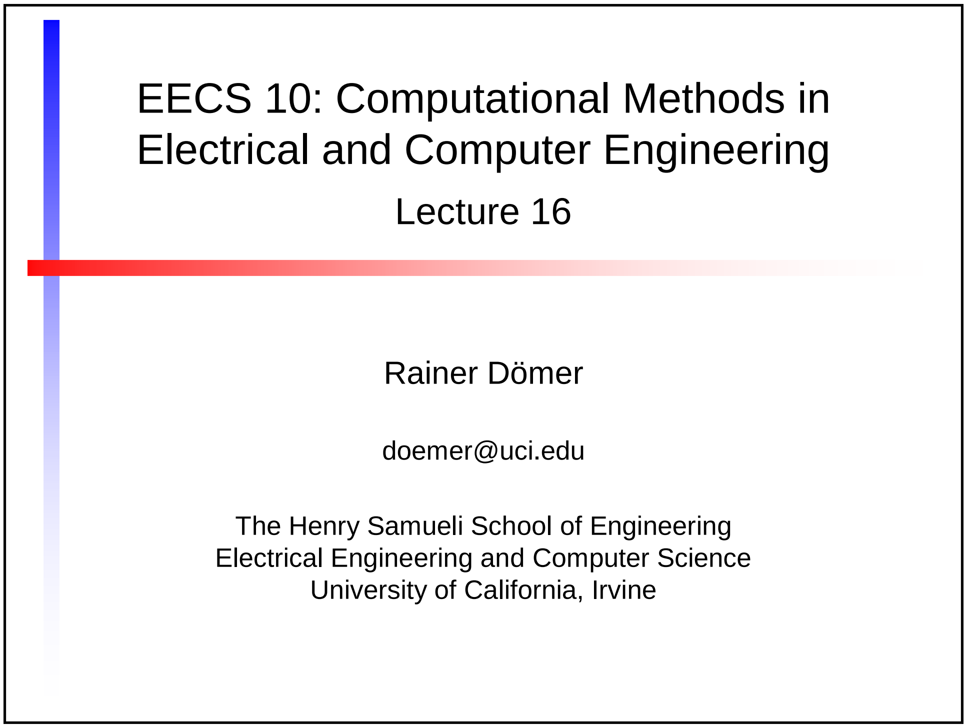


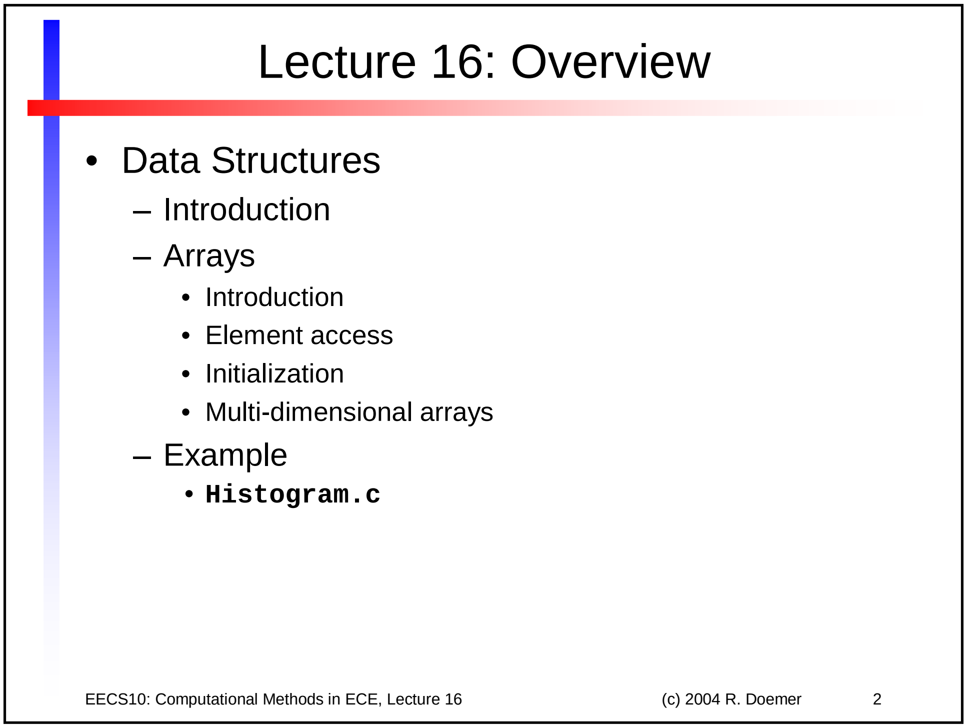
EECS 10: Computational Methods in Electrical and Computer Engineering

Lecture 16

Rainer Dömer

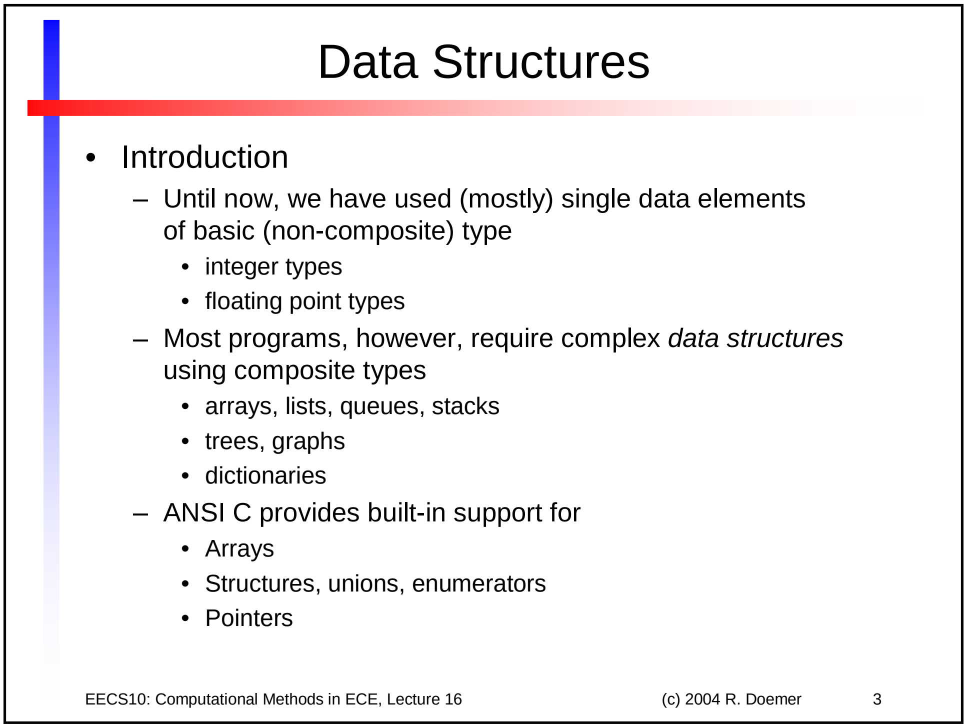
doemer@uci.edu

The Henry Samueli School of Engineering
Electrical Engineering and Computer Science
University of California, Irvine



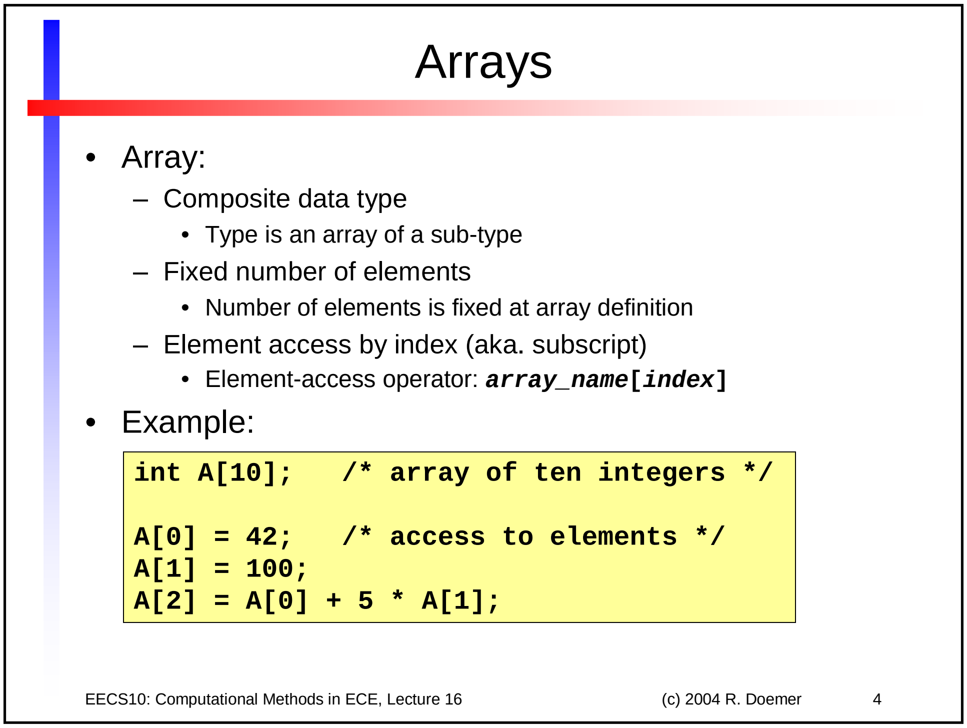
Lecture 16: Overview

- Data Structures
  - Introduction
  - Arrays
    - Introduction
    - Element access
    - Initialization
    - Multi-dimensional arrays
  - Example
    - **`Histogram.c`**



Data Structures

- Introduction
  - Until now, we have used (mostly) single data elements of basic (non-composite) type
    - integer types
    - floating point types
  - Most programs, however, require complex *data structures* using composite types
    - arrays, lists, queues, stacks
    - trees, graphs
    - dictionaries
  - ANSI C provides built-in support for
    - Arrays
    - Structures, unions, enumerators
    - Pointers

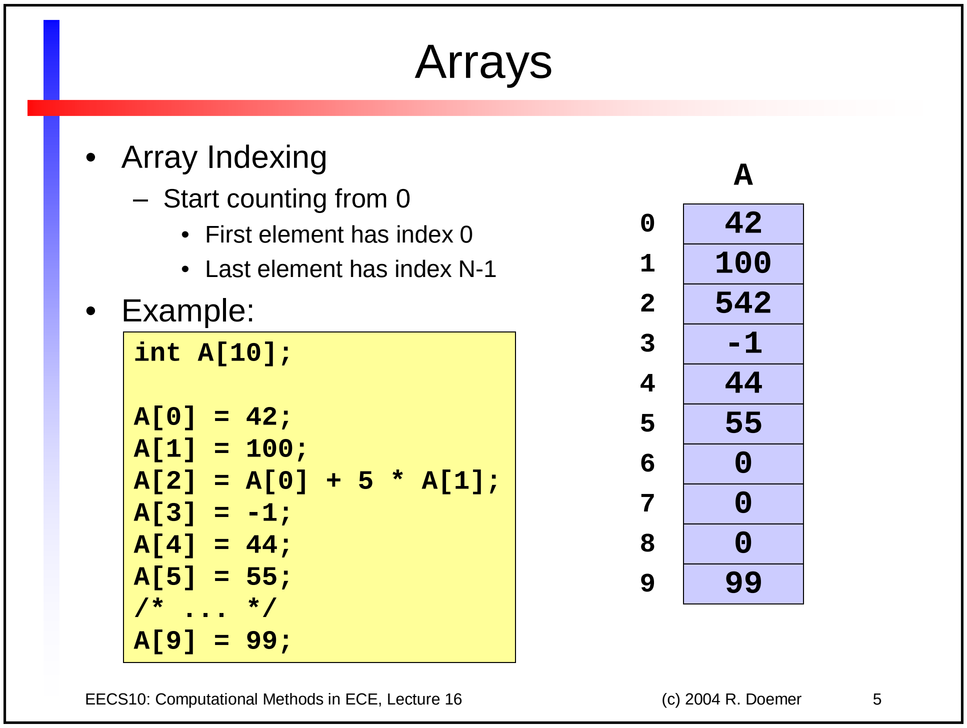


Arrays

- Array:
  - Composite data type
    - Type is an array of a sub-type
  - Fixed number of elements
    - Number of elements is fixed at array definition
  - Element access by index (aka. subscript)
    - Element-access operator: ***array_name*****[*****index*****]**
- Example:

```
int A[10];   /* array of ten integers */

A[0] = 42;   /* access to elements */
A[1] = 100;
A[2] = A[0] + 5 * A[1];
```



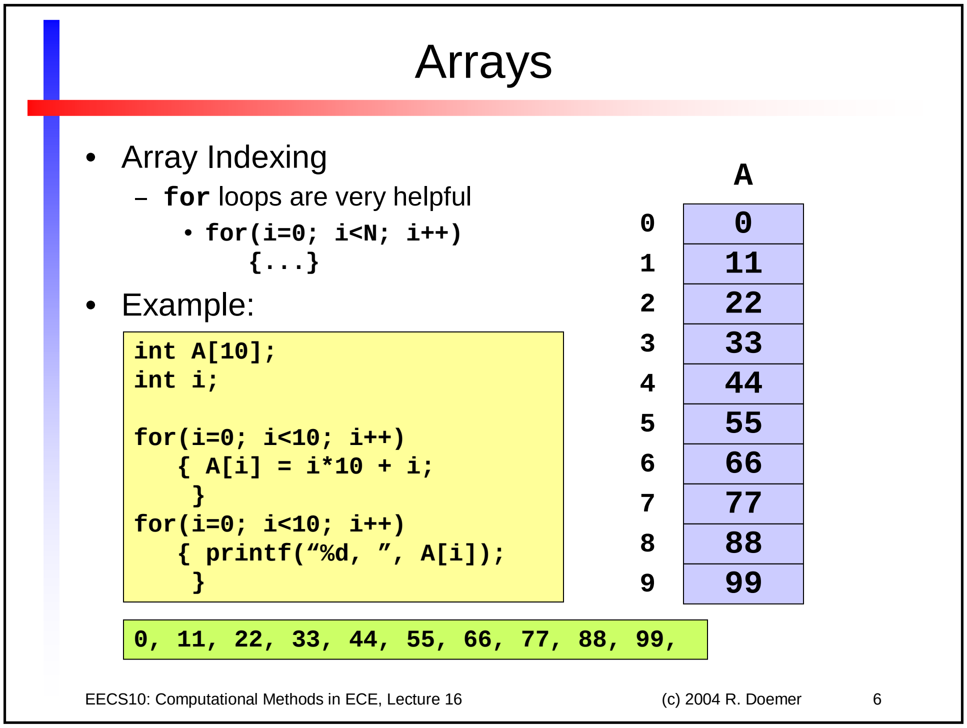
Arrays

- Array Indexing
  - Start counting from 0
    - First element has index 0
    - Last element has index N-1
- Example:

```
int A[10];

A[0] = 42;
A[1] = 100;
A[2] = A[0] + 5 * A[1];
A[3] = -1;
A[4] = 44;
A[5] = 55;
/* ... */
A[9] = 99;
```

	A
0	42
1	100
2	542
3	-1
4	44
5	55
6	0
7	0
8	0
9	99



Arrays

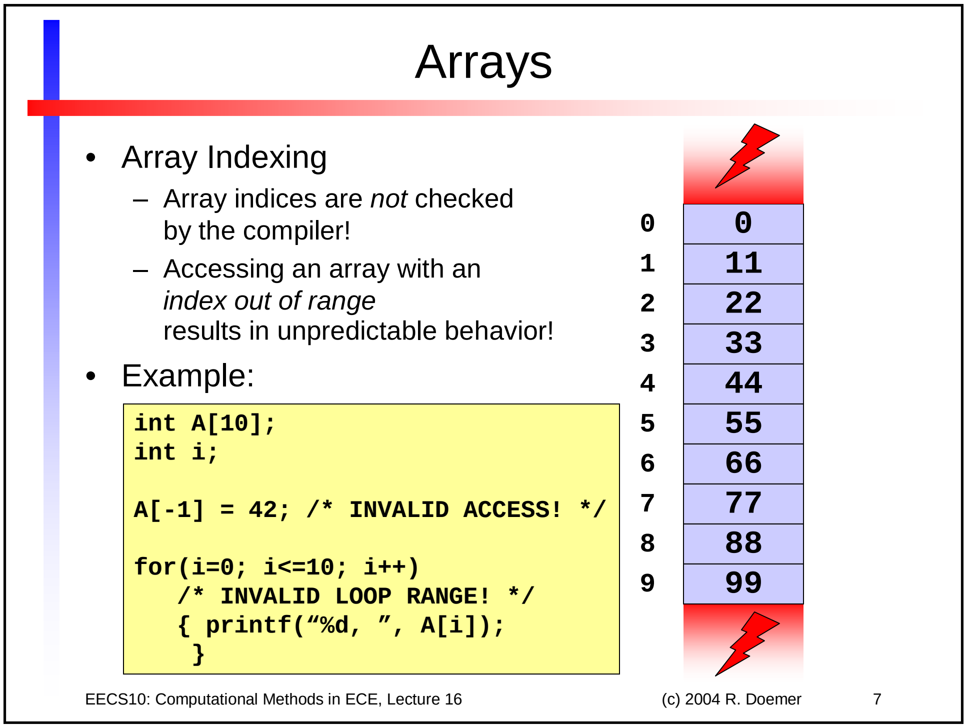
- Array Indexing
  - **for** loops are very helpful
    - **for(i=0; i<N; i++) {...}**
- Example:

```
int A[10];
int i;

for(i=0; i<10; i++)
   { A[i] = i*10 + i;
    }
for(i=0; i<10; i++)
   { printf(“%d, ”, A[i]);
    }
```

```
0, 11, 22, 33, 44, 55, 66, 77, 88, 99,
```

	A
0	0
1	11
2	22
3	33
4	44
5	55
6	66
7	77
8	88
9	99



Arrays

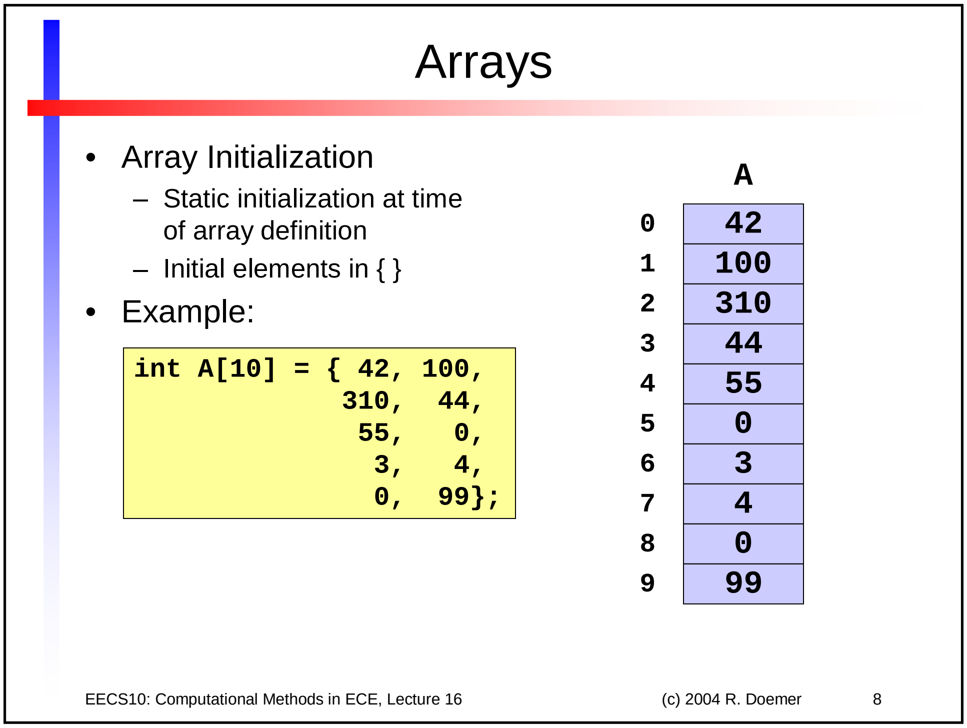
- Array Indexing
  - Array indices are *not* checked by the compiler!
  - Accessing an array with an *index out of range* results in unpredictable behavior!
- Example:

```
int A[10];
int i;

A[-1] = 42; /* INVALID ACCESS! */

for(i=0; i<=10; i++)
   /* INVALID LOOP RANGE! */
   { printf(“%d, ”, A[i]);
    }
```

0	0
1	11
2	22
3	33
4	44
5	55
6	66
7	77
8	88
9	99

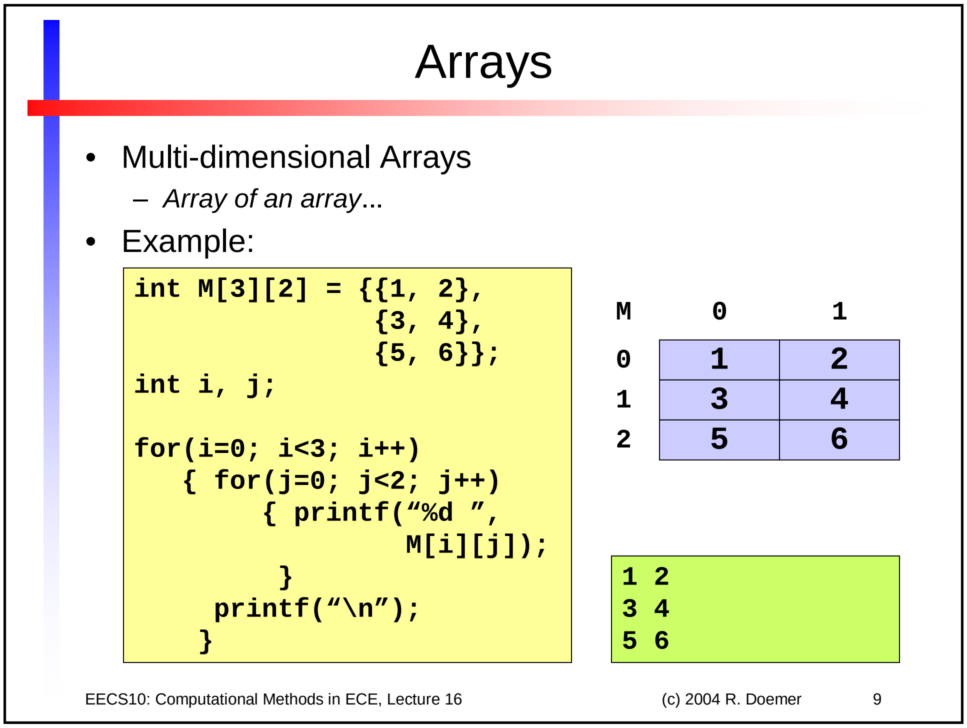


Arrays

- Array Initialization
  - Static initialization at time of array definition
  - Initial elements in { }
- Example:

```
int A[10] = { 42, 100,
             310,  44,
              55,   0,
               3,   4,
               0,  99};
```

	A
0	42
1	100
2	310
3	44
4	55
5	0
6	3
7	4
8	0
9	99



Arrays

- Multi-dimensional Arrays
  - *Array of an array*...
- Example:

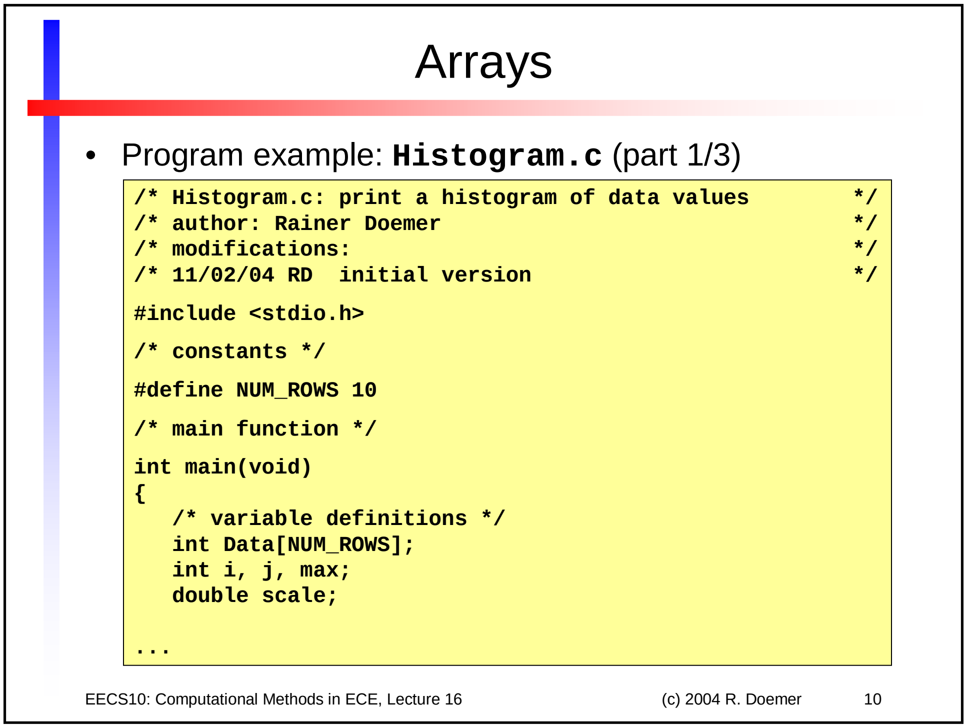
```
int M[3][2] = {{1, 2},
               {3, 4},
               {5, 6}};
int i, j;

for(i=0; i<3; i++)
   { for(j=0; j<2; j++)
        { printf(“%d ”,
                 M[i][j]);
         }
     printf(“\n”);
    }
```

M	0	1
0	1	2
1	3	4
2	5	6

```
1 2
3 4
5 6
```

EECS10: Computational Methods in ECE, Lecture 16   (c) 2004 R. Doemer   9



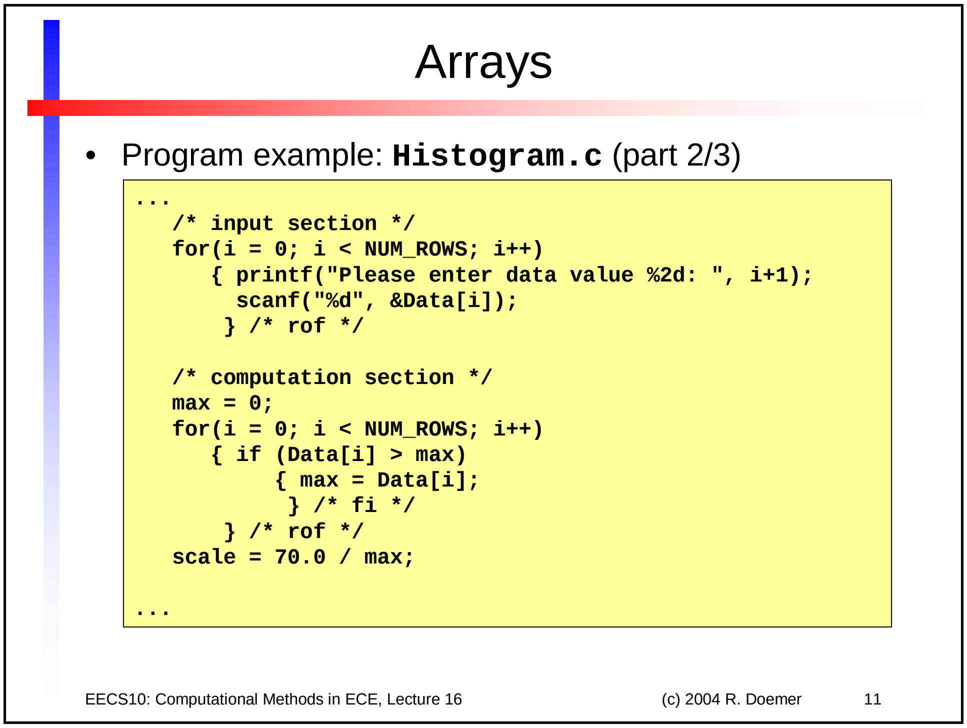
Arrays

- Program example: **`Histogram.c`** (part 1/3)

```
/* Histogram.c: print a histogram of data values        */
/* author: Rainer Doemer                                */
/* modifications:                                       */
/* 11/02/04 RD  initial version                         */
#include <stdio.h>
/* constants */
#define NUM_ROWS 10
/* main function */
int main(void)
{
   /* variable definitions */
   int Data[NUM_ROWS];
   int i, j, max;
   double scale;

...
```

EECS10: Computational Methods in ECE, Lecture 16   (c) 2004 R. Doemer   10



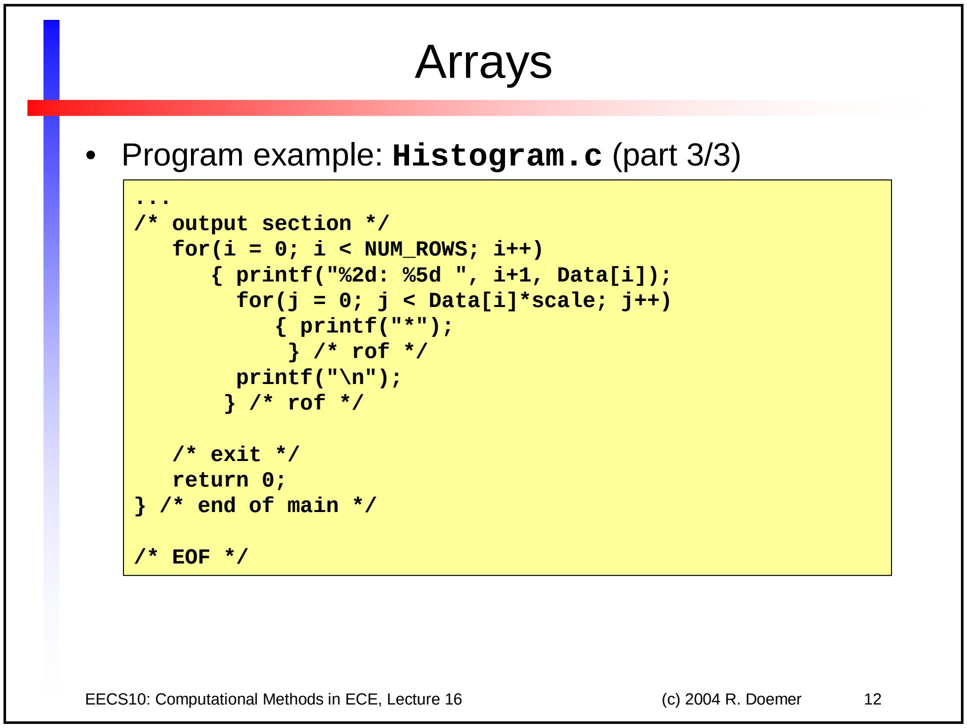
Arrays

- Program example: **`Histogram.c`** (part 2/3)

```
...
   /* input section */
   for(i = 0; i < NUM_ROWS; i++)
      { printf("Please enter data value %2d: ", i+1);
        scanf("%d", &Data[i]);
       } /* rof */

   /* computation section */
   max = 0;
   for(i = 0; i < NUM_ROWS; i++)
      { if (Data[i] > max)
           { max = Data[i];
            } /* fi */
       } /* rof */
   scale = 70.0 / max;

...
```



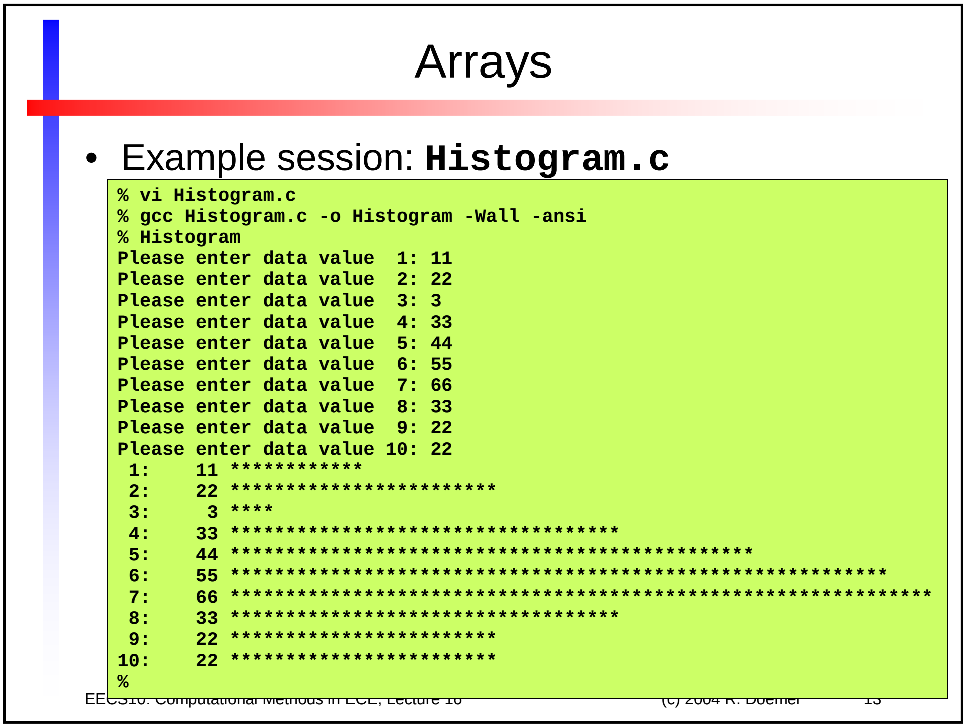
Arrays

- Program example: **`Histogram.c`** (part 3/3)

```
...
/* output section */
   for(i = 0; i < NUM_ROWS; i++)
      { printf("%2d: %5d ", i+1, Data[i]);
        for(j = 0; j < Data[i]*scale; j++)
           { printf("*");
            } /* rof */
        printf("\n");
       } /* rof */

   /* exit */
   return 0;
} /* end of main */

/* EOF */
```



Arrays

- Example session: **`Histogram.c`**

```
% vi Histogram.c
% gcc Histogram.c -o Histogram -Wall -ansi
% Histogram
Please enter data value  1: 11
Please enter data value  2: 22
Please enter data value  3: 3
Please enter data value  4: 33
Please enter data value  5: 44
Please enter data value  6: 55
Please enter data value  7: 66
Please enter data value  8: 33
Please enter data value  9: 22
Please enter data value 10: 22
 1:    11 ************
 2:    22 ************************
 3:     3 ****
 4:    33 ***********************************
 5:    44 ***********************************************
 6:    55 ***********************************************************
 7:    66 ***************************************************************
 8:    33 ***********************************
 9:    22 ************************
10:    22 ************************
%
```