

Introduction to Programming in C

(IS-FEE-10061S)

Białystok University of Technology
Faculty of Electrical Engineering
Academic year 2023/2024

Workshop no. 07 (11.04.2024)

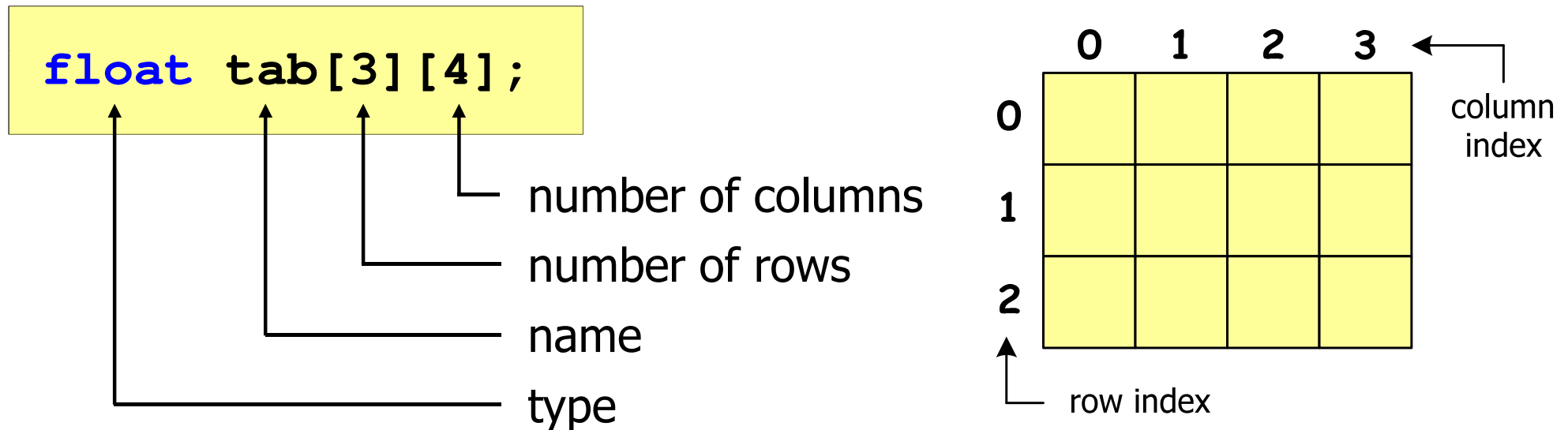
Jarosław Forenc, PhD

Topics

- Two-dimensional arrays in C
 - declaration
 - accessing array elements
 - initializing elements
 - operations

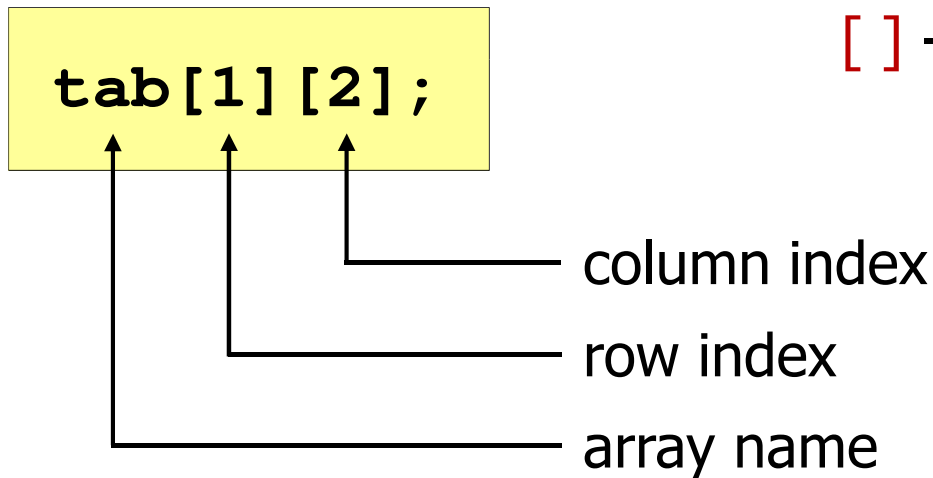
- Multidimensional arrays in C

Two-dimensional array: declaration

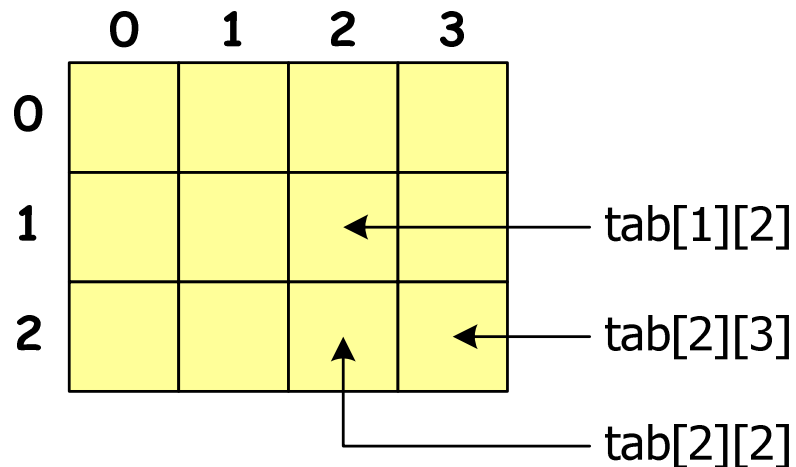


- Array **size** (number of rows, number of columns) is the value:
 - integer, positive
 - known at the compilation stage
(number: **5**, `#define N 5`, `const int n = 5;`)

Two-dimensional array: accessing array elements



[] - indexing operator



- Index:
 - number, e.g. 0, 1, 10
 - variable, e.g. i, idx
 - expression, e.g. i*j+5
- No validation of the indexes!

Two-dimensional array: initializing elements

```
int T[2][3] = {{1, 2, 3}, {4, 5, 6}};
```

	0	1	2
0	1	2	3
1	4	5	6

```
int T[2][3] = {1, 2, 3, 4, 5, 6};
```

	0	1	2
0	1	2	3
1	4	0	0

```
int T[2][3] = {1, 2, 3, 4};
```

	0	1	2
0	1	0	0
1	4	5	0

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

Two-dimensional array: initializing elements

```
int T[2][3] = {0};
```

```
int T[2][3] = {};
```

writing zeros to the array

	0	1	2
0	0	0	0
1	0	0	0

```
int T[][3] = {{1, 2, 3}, {4, 5, 6}};
```

omitting the number of rows

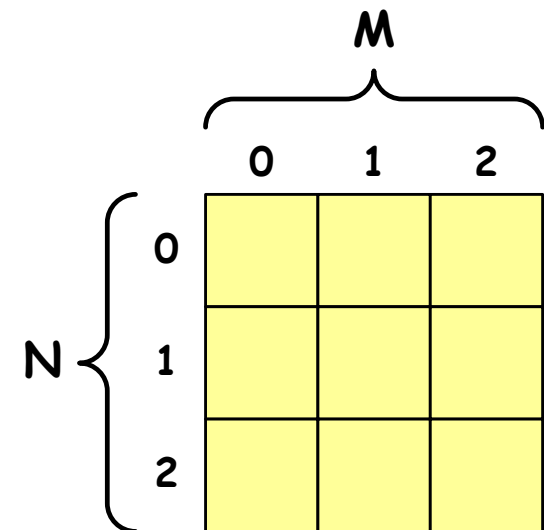
	0	1	2
0	1	2	3
1	4	5	6

Two-dimensional array: operations

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

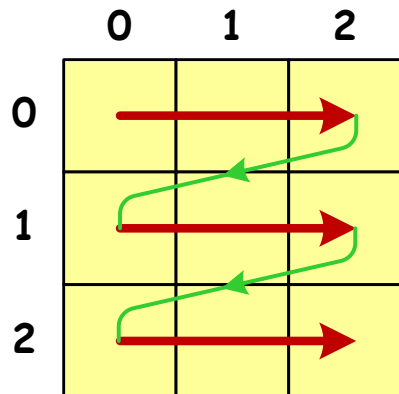
#define N 3      /* number of rows */
#define M 3      /* number of columns */

int main(void)
{
    int  tab[N][M];
    int  i, j;
```



Two-dimensional array: operations

```
/* generating array elements */  
  
srand((unsigned int) time(NULL));  
for (i=0; i<N; i++)  
    for (j=0; j<M; j++)  
        tab[i][j] = rand() % 10;
```



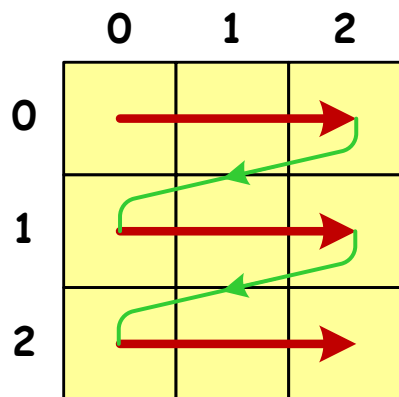
the order in which
the values of the array
elements are written

		M		
		0	1	2
N	0	9	3	1
	1	6	4	8
	2	9	4	6

Two-dimensional array: operations

```
/* printing array elements */  
  
for (i=0; i<N; i++)  
{  
    for (j=0; j<M; j++)  
        printf("%3d", tab[i][j]);  
    printf("\n");  
}
```

9	3	1
6	4	8
9	4	6

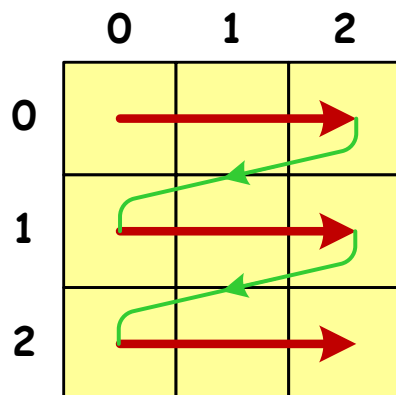


	0	1	2
0	9	3	1
1	6	4	8
2	9	4	6

Two-dimensional array: operations

```
/* finding the element with the minimum value */  
  
int min = tab[0][0];  
for (i=0; i<N; i++)  
    for (j=0; j<M; j++)  
        if (tab[i][j] < min)  
            min = tab[i][j];  
printf("Minimum: %d\n", min);
```

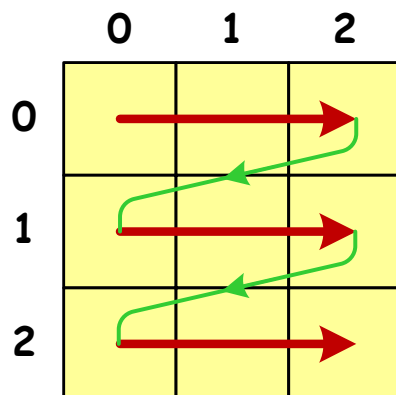
Minimum: 1



	0	1	2
0	9	3	1
1	6	4	8
2	9	4	6

Two-dimensional array: operations

```
/* sum and arithmetic mean of array elements */  
  
int sum = 0;  
for (i=0; i<N; i++)  
    for (j=0; j<M; j++)  
        sum = sum + tab[i][j];  
float mean = (float) sum / (N*M);  
printf("Sum: %d\n", sum);  
printf("Mean: %f\n\n", mean);
```



	0	1	2
0	9	3	1
1	6	4	8
2	9	4	6

Sum: 50
Mean: 5.555555

Two-dimensional array: operations

```
/* sum of the elements in the rows */  
  
for (i=0; i<N; i++)  
{  
    sum = 0;  
    for (j=0; j<M; j++)  
        sum = sum + tab[i][j];  
    printf("Sum of row %d = %d\n", i, sum);  
}
```

	0	1	2
0			
1			
2			

	0	1	2
0	9	3	1
1	6	4	8
2	9	4	6

Sum of row 0 = 13
Sum of row 1 = 18
Sum of row 2 = 19

Two-dimensional array: operations

```
/* sum of the elements in the columns */  
  
for (j=0; j<M; j++)  
{  
    sum = 0;  
    for (i=0; i<N; i++)  
        sum = sum + tab[i][j];  
    printf("Sum of column %d = %d\n", j, sum);  
}
```

	0	1	2
0			
1			
2			

	0	1	2
0	9	3	1
1	6	4	8
2	9	4	6

Sum of column 0 = 24
Sum of column 1 = 11
Sum of column 2 = 15

Multidimensional array

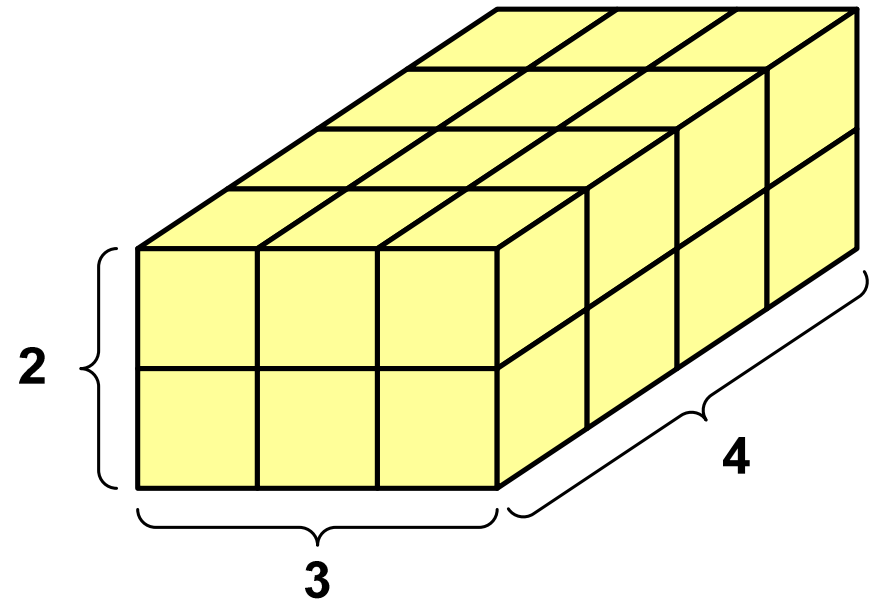
- Multidimensional array declaration

type name [dim_1] [dim_2]...[dim_N]

- Declaration of a three-dimensional array

```
int tab[4][2][3];
```

- Initializing and accessing elements is analogous to two-dimensional arrays



Multidimensional array

```
#include <stdio.h>
```

```
#define X 3
```

```
#define Y 2
```

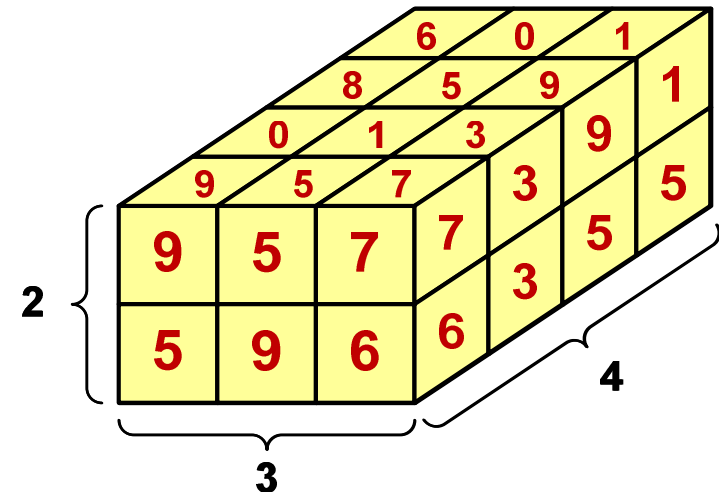
```
#define Z 4
```

```
int main(void)
```

```
{
```

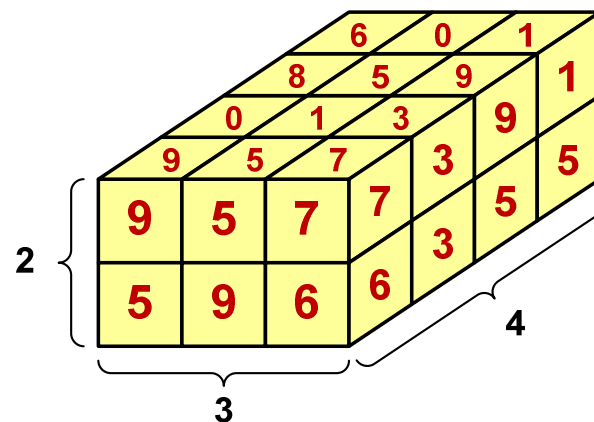
```
    int x, y, z;
```

```
    int tab[Z][Y][X] = {{{9, 5, 7}, {5, 9, 6}},  
                        {{0, 1, 3}, {7, 4, 3}},  
                        {{8, 5, 9}, {1, 3, 5}},  
                        {{6, 0, 1}, {8, 2, 5}}};
```



Multidimensional array

```
for (z=0; z<Z; z++)  
{  
    for (y=0; y<Y; y++)  
    {  
        for (x=0; x<X; x++)  
            printf("%3d", tab[z][y][x]);  
        printf("\n");  
    }  
    printf("\n");  
}  
  
return 0;  
}
```



9	5	7
5	9	6
0	1	3
7	4	3
8	5	9
1	3	5
6	0	1
8	2	5

End of workshop no. 07

Thank you for your attention!