

Fundamentals of Computing: Lecture 17

Piyush P Kurur
Office no: 224
Dept. of Comp. Sci. and Engg.
IIT Kanpur

September 7, 2009

Summary of the previous lecture

- ▶ Arrays are passed to functions as pointers.
- ▶ A function that expects an array of type `T` should have as argument type `T *`.
- ▶ There is no way a function can know the length of the array passed unless one of the argument is itself the length. This is unlike arrays in Java.

Pointers to arrays and array of pointers

```
int a[2][3], *b[3], (*c)[3];
```

Pointers to arrays and array of pointers

```
int a[2][3], *b[3], (*c)[3];
```

- ▶ a is a 2 dimensional array integer.
- ▶ b is an array 3 of pointers to integers.
- ▶ c is a pointer to an array of 3 integers.

Two dimensional arrays

- ▶ A two dimensional array is declared as `T a[2][3]`.

```
int a[2][3]
```

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$

Two dimensional arrays

- ▶ A two dimensional array is declared as `T a[2][3]`.
- ▶ The array is row major, i.e. the second index moves faster.

```
int a[2][3]
```

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$

Two dimensional arrays

- ▶ A two dimensional array is declared as `T a[2][3]`.
- ▶ The array is row major, i.e. the second index moves faster.

`int a[2][3]`

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$

Actual storage in memory

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{1,0}$	$a_{1,1}$	$a_{1,2}$
-----------	-----------	-----------	-----------	-----------	-----------

Two dimensional arrays

- ▶ A two dimensional array is declared as `T a[2][3]`.
- ▶ The array is row major, i.e. the second index moves faster.
- ▶ The expression `a` is of type pointer to array of 3 ints and therefore `a` points to `a[0][0]` and `a+1` points to `a[1][0]`.

`int a[2][3]`

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$

Actual storage in memory

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{1,0}$	$a_{1,1}$	$a_{1,2}$
-----------	-----------	-----------	-----------	-----------	-----------

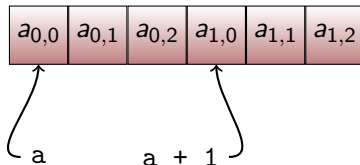
Two dimensional arrays

- ▶ A two dimensional array is declared as `T a[2][3]`.
- ▶ The array is row major, i.e. the second index moves faster.
- ▶ The expression `a` is of type pointer to array of 3 ints and therefore `a` points to `a[0][0]` and `a+1` points to `a[1][0]`.

`int a[2][3]`

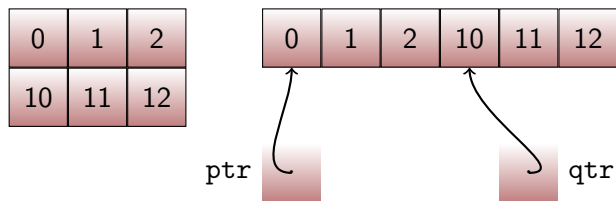
$a_{0,0}$	$a_{0,1}$	$a_{0,2}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$

Actual storage in memory



Two dimensional array and pointer to array

```
int a[2][3] = {  
                00,01,02,  
                10,11,12  
            }  
int (*ptr)[3], qtr[][3];  
ptr = a;  
qtr = a + 1;
```



Passing multi-dimensional array to function

- ▶ For a 2-d array `T a[m][n]` remember the value of `a` is pointer to an array of `n` `T`'s.
- ▶ Therefore to pass such an array declare the argument as `T (*ptr)[n]`

Example code adding matrices

```
/* adding 5x5 matrix */
int addMatrix(double (*a)[5], double (*b)[5], double (*c)
{
    for(int i = 0; i < 5; i++)
    {
        for(int j = 0; j < 5; j++)
        {
            c[i][j] = a[i][j] + b[i][j];
        }
    }
}
```

