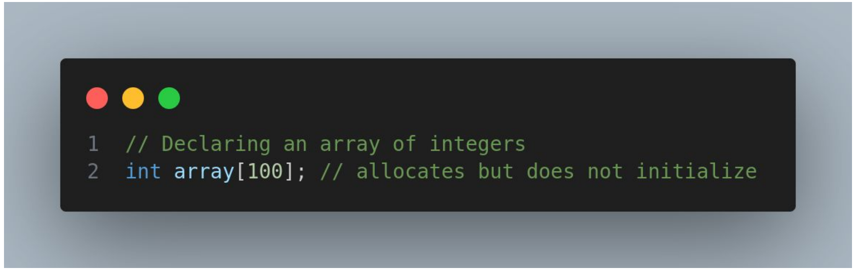
# Arrays

plus some basic data structures

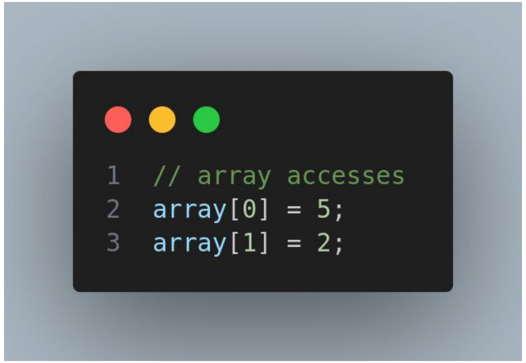
23 October 2023

UNIVERSITY OF MICHIGAN

# Quick reference for arrays

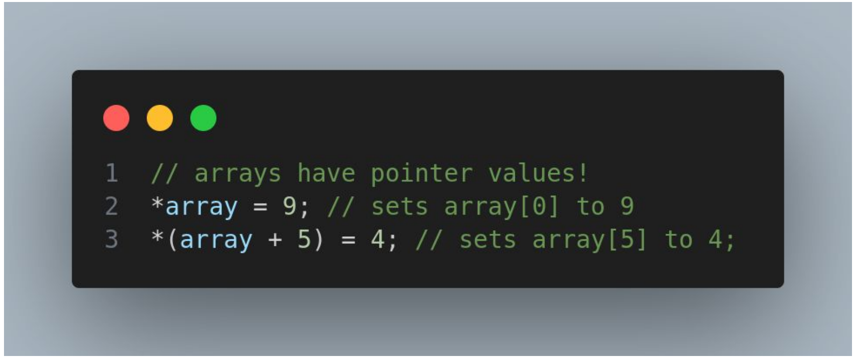


```c
// Declaring an array of integers
int array[100]; // allocates but does not initialize
```



```c
// array accesses
array[0] = 5;
array[1] = 2;
```

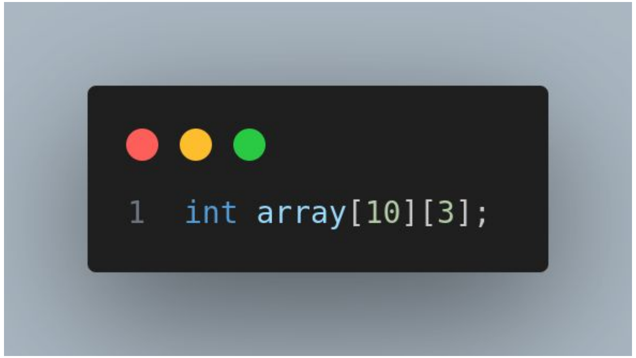
- Array variable has pointer type



```c
// arrays have pointer values!
*array = 9; // sets array[0] to 9
*(array + 5) = 4; // sets array[5] to 4;
```

# A multidimensional array is just an array of arrays

- Multidimensional: 2D array, 3D array, 4D array, ...



```
int array[10][3];
```

| | | | |
|---|---|---|---|
| array[0][0] | array[1][0] | array[2][0] | ... |
| array[0][1] | array[1][1] | array[2][1] | ... |
| array[0][2] | array[1][2] | array[2][2] | ... |

# How are multidimensional arrays laid out in memory?

- C uses row-major order
    - Some languages (Fortran, MATLAB) default to column-major

| | | | | | | |
|---|---|---|---|---|---|---|
| `array[0][0]` | `array[0][1]` | `array[0][2]` | `array[1][0]` | `array[1][1]` | `array[1][2]` | … |

- The closer together consecutive accesses are, the faster the program
    - This is a more advanced concept called *caching*
    - If you are curious about it, I recommend a Carnegie Mellon lecture here: https://www.cs.cmu.edu/afs/cs/academic/class/15213-f09/www/lectures/23-caches.pdf

# Basic data structures

## Stack and queue

23 October 2023

# Data structure: any method of organizing data

- In C / C++, usually represented by a struct / class
- Arrays are the most basic data structure
- There are many fundamental data structures you will learn:
    - Arrays
    - Stacks, queues
    - Linked lists
    - Hash tables
    - Graphs
    - Trees, tries
- For today, we will introduce stacks and queues

# Stacks are just LIFO arrays

- LIFO = “Last in, first out”
    - The **last** element you put **in** is the **first** element that goes **out**

```
// init. with empty stack
```

|   |   |   |
|---|---|---|
|   |   |   |

```
insert(1)
```

|   |   |   |
|---|---|---|
| **1** |   |   |

```
insert(2)
```

|   |   |   |
|---|---|---|
| 1 | **2** |   |

```
pop()
```

|   |   |   |
|---|---|---|
| 1 | ~~**2**~~ |   |

```
insert(5)
```

|   |   |   |
|---|---|---|
| 1 | **5** |   |

```
pop()
```

|   |   |   |
|---|---|---|
| 1 | ~~**5**~~ |   |

# Queues are just FIFO arrays

- FIFO = “First in, first out”
  - The **first** element you put **in** is the **first** element that goes **out**

```
// init. with empty stack
```

|  |  |  |
|---|---|---|
|  |  |  |

```
insert(1)
```

|  |  |  |
|---|---|---|
| **1** |  |  |

```
insert(2)
```

|  |  |  |
|---|---|---|
| 1 | **2** |  |

```
pop()
```

|  |  |  |
|---|---|---|
| ~~1~~ | 2 |  |

```
insert(5)
```

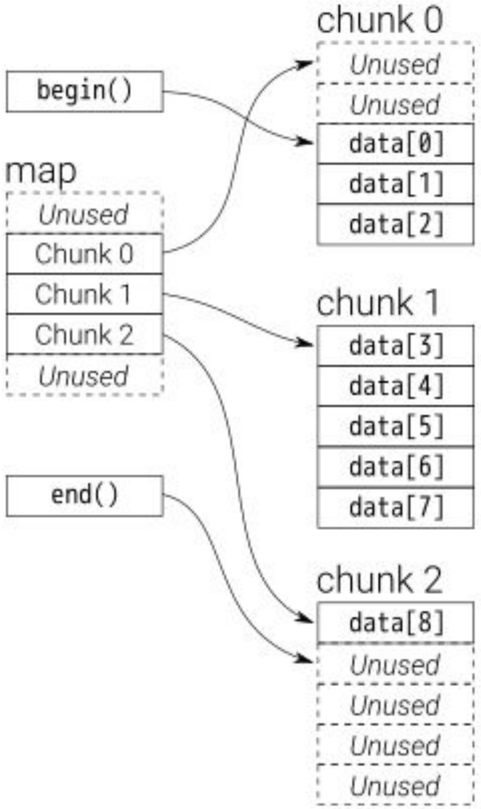
|  |  |  |
|---|---|---|
| 2 | **5** |  |

```
pop()
```

|  |  |  |
|---|---|---|
| ~~2~~ | 5 |  |

# Small aside on implementation

- At my university, we learned that the most universally efficient way to implement stacks and queues was to use a **deque**
    - "Universally efficient" is not an official term...
    - ...but I'm using it to mean that you can use the same underlying data structure for both and keep every stack and queue function efficient
- Deques are cool, read up on them on Wikipedia if you are curious :)
    - (If it looks scary, it's just pointers!)



# Sample problems

See repository for problems and solutions

# Further reading and exercises

# If you want more English-language materials...

- In C++:
    - University of Michigan: EECS 183, EECS 280
- In Python:
    - Carnegie Mellon University: CS 15-112
    - Stanford University: CS106A

- Keep in mind: C++ and Python have very different rules...
    - ...but the foundations for both languages are similar

# On your sticky note, please write…

- Your name
  - I will try to pronounce this, please forgive my Polish
- Something you enjoy
- What would you like to learn from me?
- Anything else you want to write on the sticky note

- I will collect these for attendance
  - ^ (you don't need to write this one :p)

# Contact me

email: kwang@al.edu.pl

Office hours: by appointment (send an email!)

Please reach out with any questions about assignments, computer science at an American university, the tech industry, life in the US, etcetera.

See you again soon!