PIC 10A 2B

TA: Bumsu Kim

Today…

- Vectors
- Exercises on Vectors
    - Vector Algorithms from the Lecture Slides
- HW5 Questions?

Vectors, Arrays, and Pointers

Basic Syntax

Comparisons Between Vector and Arrays // , and Arrays and Pointers

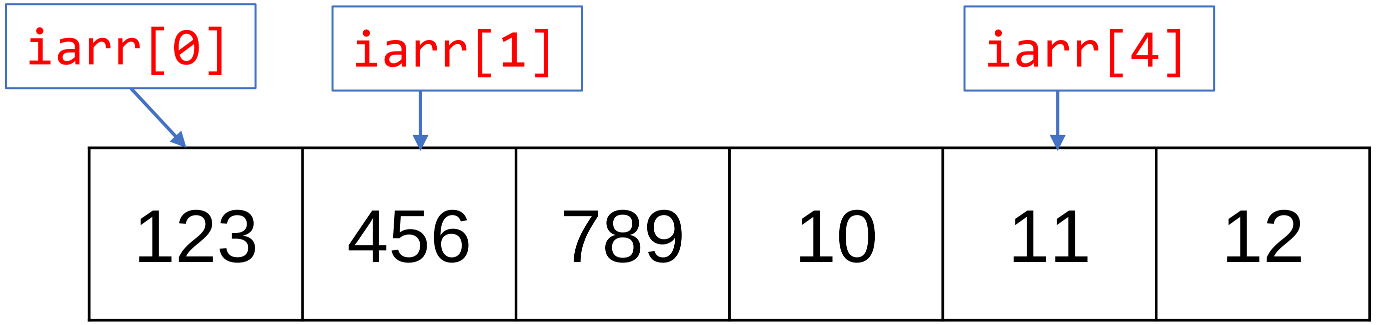
// Pointer Arithmetic

Vectors

- Vector is a special type of a class ("templated class") that can be considered an **array** of some other class type, with useful member functions
  - Recall that a "string" can be thought as an array of "char" variables, with useful member functions such as length, substr, etc.
- Must include `<vector>` library to use vectors
- Ex) Vector of `int`s
  - `vector<int> vint = { 2022, 11, 10 }; // contains 2022, 11, 10 in this order`
  - Access: subscript operator []
    - `cout << vint[0] << ", " << vint[1] << ", " << vint[2] << endl;`
- The vector class has many useful constructors
  - Creating a size "N" vector: `vector<int> vint2(N);`
  - Creating a size "N" vector and initialize all with "val": `vector<int> vint3(N, val);`
  - Creating a copy of other vector(copy constructor): `vector<int> vint4(vint);`

Vectors and Arrays

- Vectors are like arrays, but with *special features* in addition
- As mentioned last time, (static) arrays are the *most basic* “array-like” objects
- Recall that an int array of size 6 may look like (in the memory):



iarr[0] iarr[1] iarr[4]

123	456	789	10	11	12

- Accessing the k-th element of iarr → iarr[k] (subscript operator)
- A vector (internally) has an array to store the data, and support the same subscript operator
- A vector not only stores the data, but also has useful member functions for it
  - push_back, insert, erase, etc.

Vectors and Arrays

- But (static) arrays cannot change in its size, and there are no member functions for array objects
  - It's why I called it the *most basic* "array-like" type
  - No push_back, insert, erase, etc.

- There is another type of arrays, "dynamic arrays" whose size can be changed dynamically
  - But not covered in this course, you will see them again in PIC 10B
  - Also this object is pretty difficult to deal with

- Vectors are much easier to handle, so in most cases you can just use vectors
  - Vectors = Arrays + Useful Features

Vector Member Functions

- Some useful member functions of `std::vector<T>`
  - size() Returns the size (#elements) of the vector
  - front(), back() Returns the element in the front and back, respectively
  - push_back(val) Adds val at the end (and thus increases the size by 1)
  - insert(pos, val) Inserts val at the position pos (it also ++size)
  - pop_back() Removes the element at the end (--size)
  - erase(pos) Removes the element at the position pos

- Here pos must be “iterators” (covered later in this course, or PIC 10B)

Vector Algorithms

- Implement the following functions for vectors
  - copy_vec Gets a vector `from`, and copies it to another vector `to`.
  - find_vec Gets a vector `v` and an input `p`, and finds the first position of `p` in `v` (if doesn’t exists, return -1)
  - remove Gets a vector `v` and a position `pos`, and removes the data at `pos`
  - insert Gets a vector `v`, position `pos`, and an input `p`, and inserts `p` at `pos`

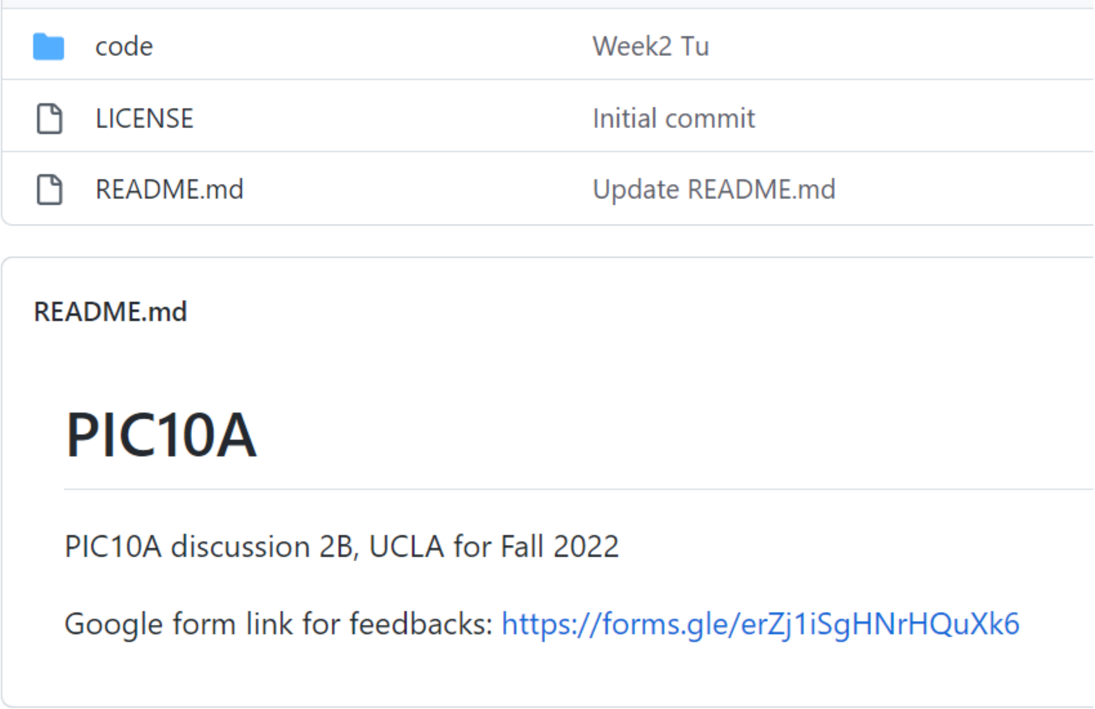
HW5 Questions?

- Or the Bonus Problem 2?

Your Feedback is welcome

- Don’t hesitate to give a feedback on the discussion
- Use the link on my Github repo, or the link below:
  - https://forms.gle/erZj1iSgHNrHQuXk6

My Github repo on the web looks like:



code	Week2 Tu
LICENSE	Initial commit
README.md	Update README.md

README.md

PIC10A

PIC10A discussion 2B, UCLA for Fall 2022

Google form link for feedbacks: https://forms.gle/erZj1iSgHNrHQuXk6

Click this link