



Lab 4

Week of 1/25



Lab Changes

- Partnering up from here on out
- One partner writes implementation, other writes test cases and tests code
- Turn in report with your test cases and the result of your tests when turning in your code
- “Sign up” with partner by writing each name on the questions sheet
- **This week**
 - One partner write *keyed_bag*, other writes *poly*
 - If you write *keyed_bag*, write test cases for *poly*, and vice versa
 - You should only be implementing *keyed_bag* OR *poly*, not both

Testing Partner's Code

- Write test code that tests each function written
 - Positive tests that should pass
 - Negative tests that should fail
 - Boundary tests that should pass
- Test code and give report to partner → partner fixes code → test code again and give report to partner [repeat as many times as necessary]
 - You need to test your partner's code at least twice and turn in report for each test

Keyed Bag

- Storing items as key, value pairs; order doesn't matter
 - Key: string
 - Value: int
- Parallel arrays
 - one to store keys
 - one to store data
- Each element stored in the bag must have a **unique** key
- Private variables
 - `key_type keys[CAPACITY];`
 - `value_type data[CAPACITY];`
 - `size_type bag_count;`

Polynomial

- Maintain a polynomial using an array
 - Index of array represents the exponent
 - Each element of the array represents the coefficient of the corresponding exponent
 - $10x^4 + 13x^3 + 31x^2 + 17x + 28$
 - `poly[4] = 10`
 - `poly[3] = 13`
 - `poly[2] = 31`
 - `poly[1] = 17`
 - `poly[0] = 28`
- Private variables
 - `double poly[MAXIMUM_DEGREE+1];`
 - `int current_degree;`

Provided Files

- Keyed Bag
 - keyed_bag.h
 - keyed_bag_tester.cpp (do not edit)
 - expected_output.txt
- Poly
 - poly.h
 - Intr_poly_tester.cpp (do not edit)
 - Polygif.cpp (do not edit)
 - Will graph polynomial and output to .gif file
- **Note:** Please name the .cpp files the same as the .h files

Demo/Testing

- Keyed Bag
 - `g++ keyed_bag.cpp keyed_bag_tester.cpp`
 - `./a.out > output.txt`
 - `diff output.txt expected_output.txt`
- Poly
 - `g++ poly.cpp intr_poly_tester.cpp`
 - `./a.out`
 - Run different commands (very similar to last week's lab)

Don't forget

- Demo code to me
 - Either today or next week
 - **Must compile and run on linux servers**
- Submit code to camino by the end of next lab
- Comment code
 - Loops and conditionals
- File with description of lab is on Camino
- Check google sheet to make sure that I didn't forget to check you off for a demo