Day 26 (Wed 03/30)

- exercise 25 review
- day 26 recap questions
- exercise 26

Announcements/reminders

- Please submit midterm project survey
  (on Gradescope) no later than Friday 4/8

- HW5: due Wednesday 4/6 by 11pm

Exercise 25

abbrev function loop

```
for ( size_t i = 0; i < word.size(); i++ ) {
  bool cur_is_vowel = is_vowel(word[i]);

  if ( cur_is_vowel ) {
    if ( !last_was_vowel ) {
      result += "";
    }
  } else {
    result += word[i];
  }

  last_was_vowel = cur_is_vowel;
}
```

Exercise 25

main function, opening input and output files

```cpp
ifstream in(argv[1]);
if ( !in.is_open() ) {
  cerr << "Couldn't open input file " << argv[1] << endl;
  return 1;
}

ofstream out(argv[2]);
if ( !out.is_open() ) {
  cerr << "Couldn't open output file " << argv[2] << endl;
  return 1;
}
```

Exercise 25

main function: main loop

```cpp
string line;
while (getline(in, line)) {
  stringstream line_in(line);
  string word;
  while (line_in >> word) {
    out << abbreviate(word) << " ";
  }
  out << endl;
}
```

Exercise 25

classify program

body of loop, variables

```
double fpval;
int ival;
string extra;
bool is_ival = false, is_fpval = false;
```

Exercise 25

classify program, body of loop

check whether token is an integer value:

```
    stringstream as_i(token);
    if (as_i >> ival) {
      if (!(as_i >> extra)) {
        sum_i += ival;
        is_ival = true;
      }
    }
```

Exercise 25

classify program, body of loop

determine whether token is a floating point value

```
if (!is_ival) {
  stringstream as_fp(token);
  if (as_fp >> fpval) {
    sum_fp += fpval;
    is_fpval = true;
  }
}
```

Exercise 25

classify program: handle tokens that are neither integer nor floating-point

```
if (!is_ival && !is_fpval) {
  ntok++;
  ntok_c += token.size();
}
```

Exercise 25

letter_freq program: initialize vector of Buckets

```
vector<Bucket> buckets;
for (int i = 0; i < 26; i++) {
  Bucket b;
  b.letter = 'a' + i;
  b.count = 0;
  buckets.push_back(b);
}
```

Exercise 25

letter_freq program: open input file, read characters, classify them

```cpp
ifstream in(argv[1]);
if (!in.is_open()) {
  cerr << "Couldn't open input file " << argv[1] << endl;
  return 1;
}

char c;
while (in.get(c)) {
  c = tolower(c);
  if (isalpha(c)) {
    buckets[c - 'a'].count++;
  }
}
```

Exercise 25

letter_freq program

Bucket comparison function

```cpp
bool compare_buckets(const Bucket &left, const Bucket &right) {
  if (left.count > right.count) { return true; }
  if (left.count < right.count) { return false; }
  return left.letter < right.letter;
}
```

Sorting the vector of Buckets:

```cpp
sort(buckets.begin(), buckets.end(), compare_buckets);
```

Exercise 25

letter_freq program: printing letter frequencies

```
 for (vector<Bucket>::const_iterator i = buckets.cbegin();
     i != buckets.cend();
     i++) {
   if (i->count > 0) {
     cout << i->letter << ": " << i->count << endl;
   }
 }
```