CS 246 Fall 2013 - Tutorial 3

September 27, 2013

1 Topics

- Strings
- Filestreams
- Constants and References

2 Fill in the Blanks - `numberFilter.cpp`

Problem: Read the first n lines of a provided filename and write each number in that file that is divisible by a given number to a provided output file.
The program has the following input:

```
> ./numberFilter
n
inFile
10
outFile
```

Note: Multiple numbers can occur on a single line. Assume all numbers are integers. Assume there will not be more than 100 numbers. Assume there will not be more than 10 lines. No error checking of input is required.
We can break this problem into 5 subproblems:

Step 1. Write the mainline program. Get the 4 arguments from standard in, open the filestreams, call appropriate functions.

Step 2. Read the first n lines of the provided input file.

Step 3. Read numbers from each the lines read and update the count of numbers seen.

Step 4. Filter the numbers by divisibility by the provided divisor and update the number of divisible integers.

Step 5. Output the divisible numbers to the appropriate file.

Step 1 has been done for you.

```
#include <string>
#include <fstream>
#include <sstream>
#include <iostream>
using namespace std;

/// Step 2.
void readLines(const int numLines, ifstream &file, string lines[]) {
  // Fill in this procedure
}

// Step 3.
void getNumbers( int *numNumbers, const int numLines, string lines[], int numbers[]) {
  // Fill in this procedure
}
```

```
// Step 4.
void filterNumbers(const int numNumbers, const int numbers[], const int divisor,
                   int filteredNumbers[], int *numFilteredNumbers) {
  // Fill in this procedure
}

// Step 5.
void outputFilteredNumbers( const int numFilteredNumbers, ofstream &outfile, const int filteredNumbers[]) {
  // Fill in this procedure
}

const int maxLines = 10;
const int maxNums = 100;
int main() {
  int numLines;
  string fileName;
  int divisor;
  string outfileName;

  cin >> numLines;
  cin >> fileName;
  cin >> divisor;
  cin >> outfileName;

  string lines[maxLines];
  int numbers[maxNum];
  int numNumbers = 0;
  int filteredNumbers[100];
  int numFilteredNumbers = 0;

  // Open Filestreams
  ifstream file(fileName.c_str()); //assume file exists
  ofstream outfile(outfileName.c_str()); //assume file exists

  // Call procedures
  readLines( numLines, file, lines);
  getNumbers(numNumbers, numLines, lines, numbers);
  filterNumbers(numNumbers, numbers, divisor, filteredNumbers, numFilteredNumbers);
  outputFilteredNumbers(numFilteredNumbers, outfile, filteredNumbers);
}
```

2.1 Testing

What would be some good tests to include in a test suite for this problem?

Answers:

- Think of some tests for discussion in tutorial.