Introduction to Systems Programming (Systems I)

Homework #3 (Part A)

Due: Wednesday February 20 2019 before 11:59 PM

Email-based help Cutoff: 5:00 PM on Tue, Feb 19 2019

Maximum Points: 15

Submission Instructions

This part of the homework assignment must be turned-in electronically via Canvas. Ensure you name this document *MUid*_*homework3*_*PartA*.docx, where *MUid* is your Miami University unique ID. Complete the method shown for each problem. For each method, you can develop and test them in NetBeans and just copy-paste your solutions into this document.

Once you have completed answering the questions save this document as a PDF file (don't just rename the document; that is not the correct way to save as PDF) and upload it to Canvas

General Note: Upload each file associated with homework (or lab exercises) individually to Canvas. Do not upload archive file formats such as zip/tar/gz/7zip/rar etc.

Objective

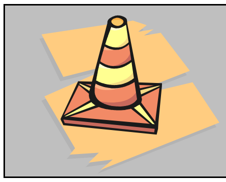
The objective of this homework is to:

- Understand working with std::vector.
- Practice answering exam style questions

Required reading

Prior to answering the questions in this homework briefly review the following chapters from the E-book titled "C++ How to Program" (all students have free access to the electronic book):

- Chapter 7.10 (std::vector)
- Chapter 14.1 – 14.6 (File I/O)



Although the Safari E-books are available to all students there are only a limited number of concurrent licenses to access the books. Consequently, do not procrastinate working on this homework or you may not be able to access the E-books due to other users using them.

1. What is quoted text and how do you read quoted text in C++? Explain with a suitable example (other than the one shown in Chapter 14) [**1 points**]

Quote text is test in a file that is surround by quotation marks. In order to read quoted text you can use the stream manipulator **quoted(),** for example:
inFile >> person >> quoted(name)
This reads the a name in quotes like “Leann Hunter” as one string and stores it in the variable **person** without the quotation marks.

2. Assume you have a method called `processLines(std::istream& is, std::ostream& os)` that process line-by-line. Complete the `main` method below to call `processLines` method to process the 3 lines: `"Line #1"`, `"Second Line"`, and `"Last line"`. The output should be written to standard console output stream. (Hint: Use a `std::istringstream`) [**3 points,** In exam you would have 5–7 minutes to write the 3 line solution]

```
// Prototype declaration
void processLines(std::istream& is, std::ostream& os);

int main() {
std::istringstream input(std::getline(std::cin, line));
process(input, std::cout);
}
```

3. Complete the following method that returns a vector with only even values in the `src` vector. If the `src` vector has values {2, -4, 7, 9, 3, 8} this method should return a vector with values {2, -4, 8}. [**3 points**, In exam you will have about 7 minutes to write the solution]

```
using IntVec = std::vector<int>;

IntVec evens(const IntVec& src) {
  IntVec src2;
  For (size_t I = 0; (  I <= src.size(); i++) {
        If(src[i] % 2 == 0) {
          Src2[i] = src[i];
        }
}
Return src2;



}
```

4. Complete the following method that returns a vector that contains a reverse of the words. For example if `src` is {`"one"`, `"two"`, `"three"`} the method should return a vector with strings {`"three"`, `"two"`, `"one"`} [**3 points**]

```
using StrVec = std::vector<std::string>;

StrVec reverse(const StrVec& src) {
  StrVec src2;
  for (size_ti = src.size() - 1; (i >= 0); i--) {
      src2[i] = src[i];
  }
Return src2;




}
```

5. Complete the following method that returns a vector with the first `n` prime numbers. For example, if `n == 7`, this method should return a vector with values {1, 2, 3, 5, 7, 11, 13}. [**5 points**, In exam you will have about 10 minutes to write the solution]

```
using IntVec = std::vector<int>;
IntVec getPrimes(int n) {
  IntVec  test;
  For(size_t I = 0; (I <= n); i++) {
     If(I % 2 == 0 && I != 0) {
      Test[i] = i;
    }
  Return test;


}
```