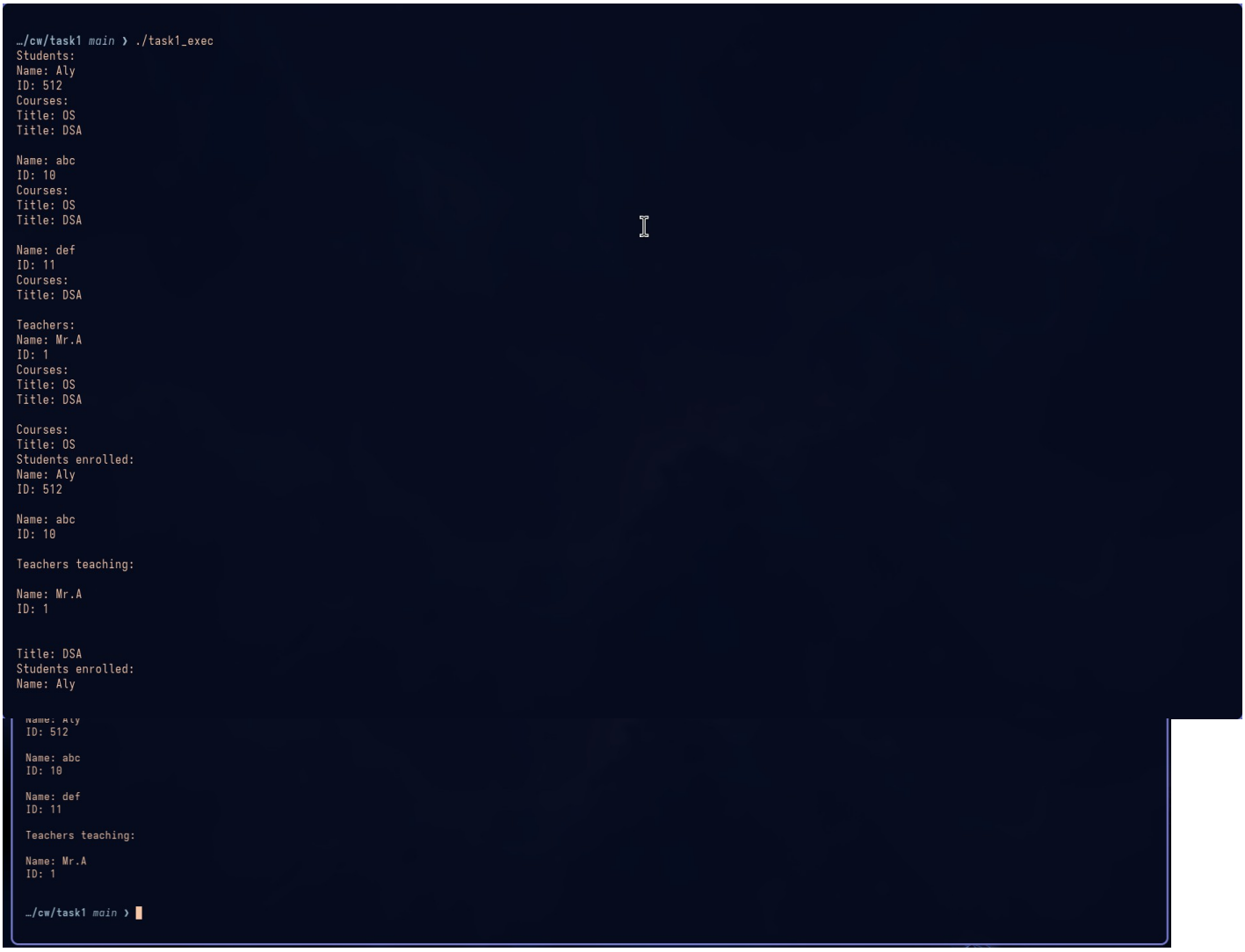
Aly Muhammad Kazani 24K-0512

OS Lab 2 Tasks

CW

task 1)



Makefile:

```
o_files = build/main.o build/student.o build/teacher.o build/course.o
output = task1_exec

# how this works:
# make build > adds all .o files to build/ and builds an executable
# make run > runs the executable, if available, else makes it first
# make clean > removes all .o files
# make full_clean > resets the directory
```

```
.PHONY: run build clean full_clean # not files

build: $(output)

$(output): $(o_files)
	g++ $(o_files) -o $(output)

build/%.o: %.cpp
	mkdir -p build
	g++ -c $< -o $@

run: $(output)
	./$(output)

clean:
	rm -rf build

full_clean: clean
	rm -f $(output)
```

Headers:

// student.h

```
#pragma once

#include <iostream>
using namespace std;

#include "course.h"

class Course;

class Student
{
private:
    const string name;
```

```
    const int id;
    vector<Course*> enrolled;
public:
    Student(const string n, const int i);
    string getName() const;
    int getID() const;

    void AddCourse(Course& c);

    void display() const;
};
```

// teacher.h

```
#pragma once

#include <iostream>
using namespace std;

#include "course.h"

class Course;

class Teacher
{
private:
    const string name;
    const int id;
    vector<Course*> teaching;
public:
    Teacher(const string name, const int id);
    string getName() const;
    int getID() const;

    void AddCourse(Course& c);
```

```
    void display() const;
};
```
// course.h
```
#pragma once

#include <iostream>
#include <string>
#include <vector>
using namespace std;

#include "student.h"
#include "teacher.h"

class Student;
class Teacher;

class Course
{
private:
    string title;
    vector<Student*> taking;
    vector<Teacher*> teaching;
public:
    Course(const string t);
    string getTitle() const;

    void AddStudent(Student& s);
    void AddTeacher(Teacher& t);

    void display() const;
};
```

Source Files:

// student.cpp
```
#include "student.h"
```

```
Student::Student(const string n, const int i) : name(n), id(i)
{

}

string Student::getName() const
{
   return name;
}

int Student::getID() const
{
   return id;
}

void Student::AddCourse(Course& c)
{
   enrolled.emplace_back(&c);
}

void Student::display() const
{
   cout << "Name: " << name << endl;
   cout << "ID: " << id << endl;

   cout << "Courses:\n";
   for(Course* c : enrolled)
   {
      cout << "Title: " << c->getTitle() << endl;
   }
}
```

// teacher.cpp

```
#include "teacher.h"
```

```
Teacher::Teacher(const string n, const int i) : name(n), id(i)
{

}

string Teacher::getName() const
{
   return name;
}

int Teacher::getID() const
{
   return id;
}

void Teacher::AddCourse(Course& c)
{
   teaching.emplace_back(&c);
}

void Teacher::display() const
{
   cout << "Name: " << name << endl;
   cout << "ID: " << id << endl;

   cout << "Courses:\n";
   for(Course* c : teaching)
   {
      cout << "Title: " << c->getTitle() << endl;
   }
}
```

// course.cpp

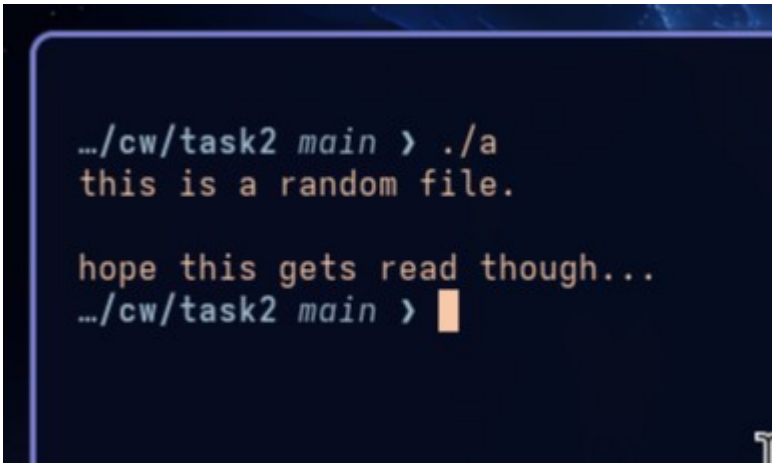
```
#include "course.h"

Course::Course(const string t) : title(t)
```

```
{

}

string Course::getTitle() const
{
   return title;
}

void Course::AddStudent(Student& s)
{
   taking.emplace_back(&s);
}

void Course::AddTeacher(Teacher& t)
{
   teaching.emplace_back(&t);
}

void Course::display() const
{
   cout << "Title: " << title << endl;

   cout << "Students enrolled:\n";
   for(auto s : taking)
   {
      cout << "Name: " << s->getName() << endl;
      cout << "ID: " << s->getID() << endl << endl;
   }

   cout << "Teachers teaching:\n" << endl;
   for(auto t : teaching)
   {
      cout << "Name: " << t->getName() << endl;
      cout << "ID: " << t->getID() << endl << endl;
```

```
    }
}
// main.cpp
#include "course.h"
#include "student.h"
#include "teacher.h"

#include <iostream>
using namespace std;

void enrollStudent(Student& s, Course& c)
{
    s.AddCourse(c);
    c.AddStudent(s);
}

void assignTeacher(Teacher& t, Course& c)
{
    t.AddCourse(c);
    c.AddTeacher(t);
}

int main()
{
    Student s1("Aly", 512);
    Student s2("abc", 10);
    Student s3("def", 11);

    Teacher t("Mr.A", 1);

    Course c1("OS");
    Course c2("DSA");

    enrollStudent(s1, c1);
    enrollStudent(s1, c2);
```

```
    enrollStudent(s2, c1);
    enrollStudent(s2, c2);

    enrollStudent(s3, c2);

    assignTeacher(t, c1);
    assignTeacher(t, c2);

    cout << "Students:" << endl;
    s1.display();
    cout << endl;
    s2.display();
    cout << endl;
    s3.display();
    cout << endl;

    cout << "Teachers:" << endl;
    t.display();
    cout << endl;

    cout << "Courses:" << endl;
    c1.display();
    cout << endl;
    c2.display();

    return 0;
}
```

task 2)

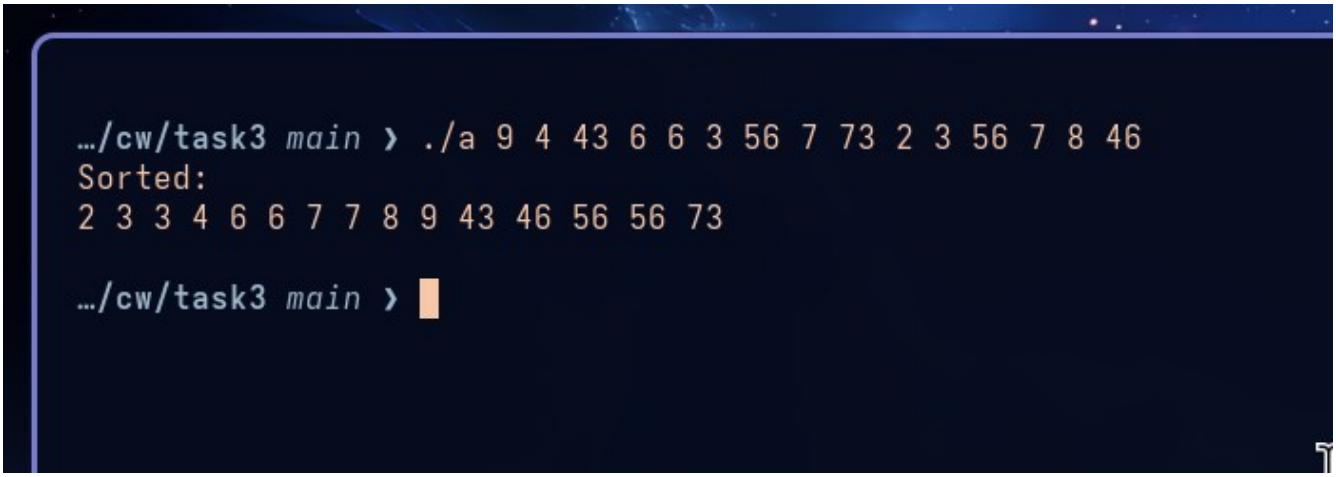



```
// file.txt
this is a random file.

hope this gets read though…
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>


int main(int argc, char** argv)
{
    const char* filename = "file.txt";
    char command[50] = "cat ";

    FILE* f = fopen(filename, "r");

    if(f == NULL)
    {
        perror("Error \'file_not_found\'");
        exit(1);
    }

    strcat(command, filename);
    system(command);
```

```
    fclose(f);

    return 0;
}
```

task 3)

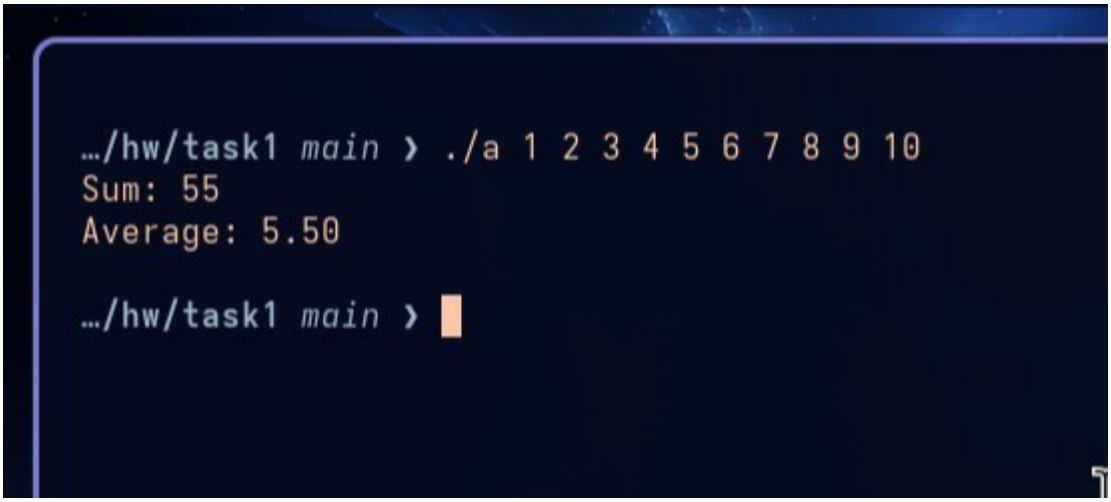



```
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

int main(int argc, char** argv)
{
    vector<int> vals;

    for(int i = 1; i < argc; i++)
    {
        vals.push_back(atoi(argv[i]));
    }

    sort(vals.begin(), vals.end());

    cout << "Sorted: " << endl;
    for(auto v : vals)
    {
        cout << v << ' ';
```

```
    }
    cout << endl;

    return 0;
}
```

HW

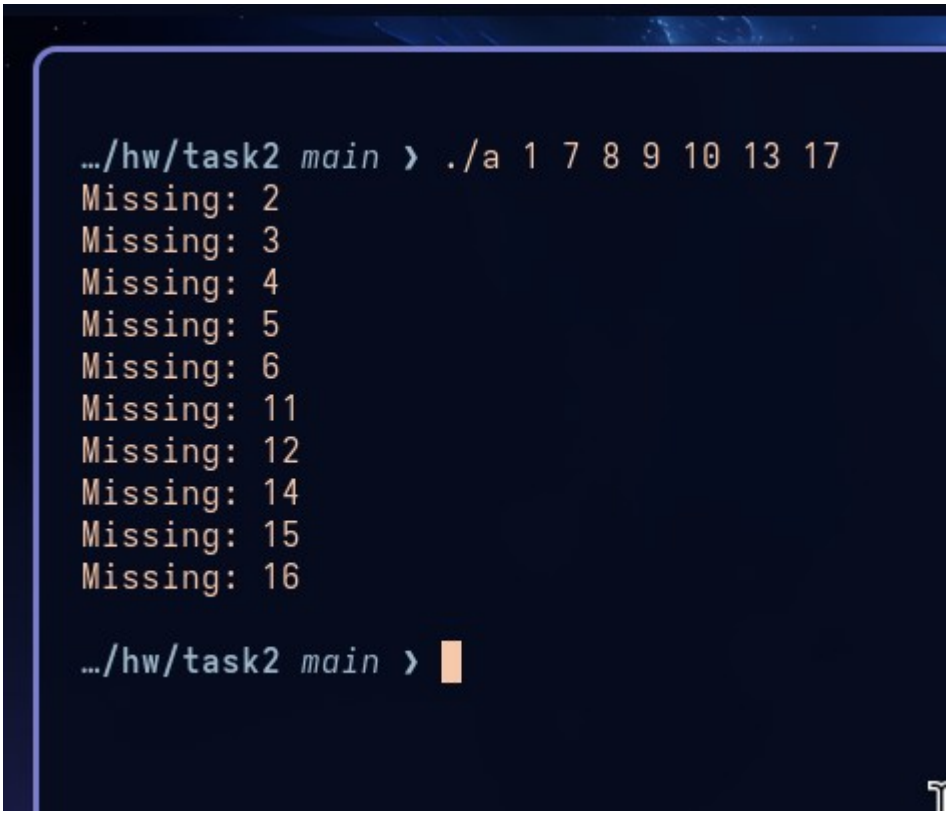
task 1)




```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char** argv)
{
        int vals[100];
        for(int i = 1; i < argc; i++)
        {
                vals[i-1] = atoi(argv[i]);
        }

        int sum = 0;
        for(int i = 0; i < argc - 1; i++)
        {
                sum += vals[i];
        }
        float avg = (float)sum / (argc - 1);
        printf("Sum: %d\nAverage: %.2f\n", sum, avg);
```

```
        return 0;
}
```

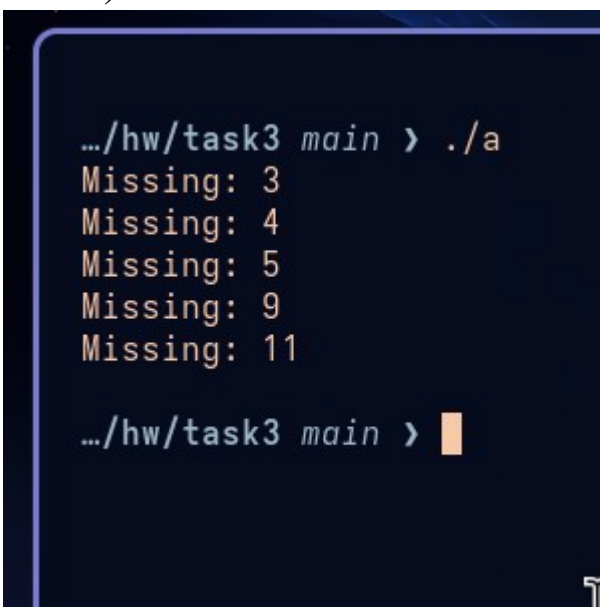
task 2)




```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char** argv)
{
    int list[100];
    for(int i = 1; i < argc; i++)
    {
        list[i - 1] = atoi(argv[i]);
    }
    int index = list[0];

    for(int i = 1; i < argc - 1; i++)
    {
```

```
        if(list[i] == index + 1)
        {
            index++;
            continue;
        }
        else
        {
            printf("Missing: %d\n", index + 1);
            index++;
            i--;
        }
    }

    return 0;
}
```

task 3)

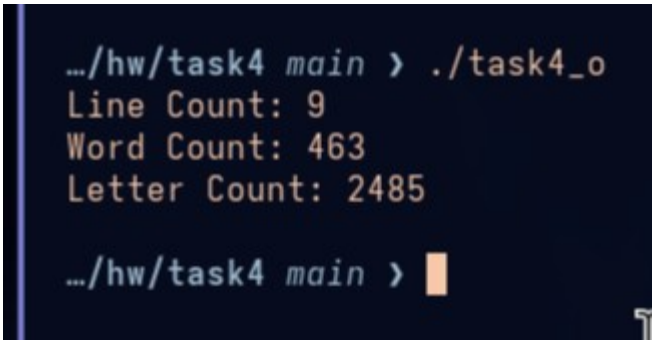



```
// seq.txt
1 2 6 7 8 10 12
// a.c
#include <stdio.h>
#include <stdlib.h>
```

```
int main() {

    FILE *file;
    char *filename = "seq.txt";
    int list[100];

    file = fopen(filename, "r");
    if(file == NULL)
    {
       printf("Error opening file!\n");
       return -1;
    }

    int index = fscanf(file, "%d", & list[0]);

    for(int i = 1; i < 100; i++)
    {
       if(fscanf(file, "%d", &list[i]) == 1)
       {
          while(list[i] != index + 1)
          {
             printf("Missing: %d\n", index + 1);
             index++;
          }
          index++;
          continue;
       }
       else
       {
          break;
       }
    }

    fclose(file);
```

```
    return 0;
}
```

task 4)



// makefile

```
src = main.cpp letter.cpp line.cpp word.cpp
o_files = build/main.o build/letter.o build/line.o build/word.o
output = task4_o

# how this works:
# make build > adds all .o files to build/ and builds an executable
# make run > runs the executable, if available, else makes it first
# make clean > removes all .o files
# make full_clean > resets the directory

.PHONY: run build clean full_clean # not files

build: $(output)

$(output): $(o_files)
	g++ $(o_files) -o $(output)

build/%.o: %.cpp
	mkdir -p build
	g++ -c $< -o $@

run: $(output)
	./$(output)
```

```
clean:
	rm -rf build

full_clean: clean
	rm -f $(output)
```

Headers:

// letter.h

```
#pragma once

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

class LetterCount
{
   private:
      int numLetters = 0;
      string buff;
   public:
      LetterCount(const string& filename);
      int getNumLetters() const;
};
```

// word.h

```
#pragma once

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

class WordCount
{
```

```
    private:
        int numWords = 0;
        string buff;
    public:
        WordCount(const string& filename);
        int getNumWords() const;
};
```

// line.h

```
#pragma once

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

class LineCount
{
    private:
        int numLines = 0;
        string buff;
    public:
        LineCount(const string& filename);
        int getNumLines() const;
};
```

Source Files:

// letter.cpp

```
#include "letter.h"

LetterCount::LetterCount(const string& filename)
{
    ifstream file(filename);

    if(file.is_open())
    {
        while(getline(file, buff))
```

```
        {
            for(char c : buff)
            {
                if(isalnum(c))
                {
                    numLetters++;
                }
            }
        }

        file.close();
    }
    else
    {
        cerr << "Unable to open file!" << endl;
    }

}

int LetterCount::getNumLetters() const
{
    return numLetters;
}
```

// word.cpp

```
#include "word.h"

WordCount::WordCount(const string& filename)
{
    ifstream file(filename);

    if(file.is_open())
    {
        while(getline(file, buff))
        {
            bool isWord = false;
```

```
            for(char c : buff)
            {
               if(isalnum(c) && !isWord)
               {
                  isWord = true;
                  numWords++;
               }
               if(c == ' ' && isWord)
               {

                  isWord = false;
               }
            }
         }
      }

      file.close();
}

int WordCount::getNumWords() const
{
   return numWords;
}
```

// line.cpp

```
#include "line.h"

LineCount::LineCount(const string& filename)
{
   ifstream file(filename);

   if(file.is_open())
   {
      while(getline(file, buff))
      {
```

```
            numLines++;
        }
    }

    file.close();
}

int LineCount::getNumLines() const
{
    return numLines;
}
```
// main.cpp
```
#include <iostream>
using namespace std;

#include "letter.h"
#include "word.h"
#include "line.h"

int main()
{
    LetterCount let("text.txt");
    LineCount li("text.txt");
    WordCount wo("text.txt");

    cout << "Line Count: " << li.getNumLines() << endl;
    cout << "Word Count: " << wo.getNumWords() << endl;
    cout << "Letter Count: " << let.getNumLetters() << endl;

    return 0;
}
```
// text.txt

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean lacinia, lacus nec vehicula tincidunt, nisi eros lobortis lectus, non consequat ipsum ligula quis quam. Maecenas ultricies ut sem sed placerat. In aliquam elit quis feugiat

aliquet. Duis magna dui, egestas non porta eget, pellentesque in neque. Interdum et malesuada fames ac ante ipsum primis in faucibus. Nunc venenatis posuere bibendum. Cras finibus magna nec nisi imperdiet consequat eget quis lacus. Etiam non nisl felis. Suspendisse ac diam et libero hendrerit ornare ut vitae justo. Vestibulum commodo felis ex, a feugiat dui vehicula vel. Vivamus eu finibus arcu. Interdum et malesuada fames ac ante ipsum primis in faucibus. Sed commodo consequat ipsum ac posuere. Mauris magna ex, cursus quis mattis vel, semper in metus. Phasellus ultrices suscipit magna sit amet mollis. Duis vel massa lectus.

Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Nullam suscipit a erat a congue. Vestibulum eu malesuada turpis. Praesent orci ante, semper ut dui non, cursus euismod risus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque luctus, ligula a pretium congue, erat dolor faucibus nisi, in varius arcu mauris non dolor. Sed posuere suscipit arcu, ut lacinia sapien. Ut efficitur tristique eros, sit amet iaculis risus gravida quis. Morbi eu volutpat felis. Phasellus id pretium justo. Suspendisse vel neque at risus interdum molestie. Morbi quis eros quam. Maecenas sed felis id orci porta lobortis. Donec facilisis orci et odio suscipit finibus. Donec placerat luctus dui, ac condimentum enim placerat a.

Nullam eros massa, efficitur at massa at, consequat mattis sem. Praesent sed sem eu nunc varius pharetra non at enim. Cras rhoncus vehicula ipsum, sed suscipit sapien sollicitudin ac. Ut sed ex ac urna egestas luctus quis non ligula. Nulla pellentesque lectus sit amet lobortis venenatis. Fusce rutrum metus dui, ac consectetur orci faucibus a. Nam diam sem, suscipit in ipsum at, egestas rhoncus lectus. Curabitur euismod pulvinar est a varius. Duis vitae lectus eu tellus auctor pulvinar vel at nisi. Proin maximus faucibus purus, vitae dapibus magna aliquam in.

Fusce ac tristique dui. Nulla at volutpat felis, sed rutrum metus. Nulla fringilla posuere semper. Praesent ut turpis libero. Donec gravida aliquet nulla sit amet pretium. Mauris vitae condimentum nisi. Curabitur eget sapien magna. Fusce feugiat pretium nibh, sit amet porta velit viverra in.

Nunc sodales consectetur blandit. Maecenas posuere, lacus sit amet condimentum sagittis, ipsum lectus accumsan mauris, quis laoreet velit felis ac metus. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. In quis orci vitae nisl sollicitudin scelerisque. Nulla nunc massa, semper sit amet dolor in, condimentum sollicitudin tellus. Praesent mattis est ullamcorper leo laoreet dapibus. Quisque velit libero, finibus ut rhoncus at, rhoncus a ligula. Sed in purus ligula. Fusce euismod diam ut dui aliquam sagittis.