**Homework 7 – CS60 Linnell**

**Arjun Kohli**

**W1579330**

**Problem 1:**

#include <iostream>

#include <vector>

#include <fstream>

#include <algorithm>

using namespace std;

int main() {

ifstream in;

ofstream out;

in.open("sample1.txt");

out.open("p1.txt");

if (in.fail() || out.fail()) {

exit(1);

}

// a

vector<string> v;

string s;

while (!in.eof()) {

in >> s;

v.push\_back(s);

}

for (auto e:v) {

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

}

for (auto e:v) {

out << e << endl;

}

//b

vector<string> v;

string s;

while (!in.eof()) {

in >> s;

v.push\_back(s);

}

for (auto e:v) {

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

}

vector<string>::iterator i;

vector<string>::iterator j;

for (i = v.begin(); i < v.end(); i++) {

for (j = v.begin()+1; j < v.end(); j++) {

if ((\*i) == (\*j)) {

v.erase(j);

}

}

}

int count = 1;

string word = v[0];

for (int i = 1; i < v.size(); i++) {

if (word != v[i]) {

out << word << " " << count << endl;

count = 0;

word = v[i];

}

count++;

}

out << word << " " << count << endl;

in.close();

out.close();

}

**Input File (sample1.txt) – The Gentle Seduction Extract**

**Output File (p1.txt) – Last 10 Lines for Question a):**

you

you

you

you

you

you

you

you

you

you

you

you

you'd

you'll

you'll

you'll

you'll

you'll

you'll

you're

you're

you're

you're

you,

you.

you.

you."

you."

you."

young

young."

younger;

your

your

your

your

your

your

your

your

your

yourself,

yourself,

yourself,

youth

**Output File (p1.txt) – Last 10 Lines for Question b):**

wisdom 1

wish 1

with 4

within 2

within. 1

without 1

woman 1

won 1

won't." 1

wonder. 1

wondered 1

wonderful. 1

word 1

words 1

wore 1

work 1

work. 1

worked 1

world 1

world, 1

worse 1

would 4

wrapped 1

wrong, 1

wrought 1

year 1

year." 1

yearning, 1

years 2

years, 1

years. 1

yet 1

yet." 1

you 3

you'll 1

you're 1

you. 1

you." 1

young." 1

your 2

yourself, 1

youth 1

**Problem 2:**

#include <iostream>

#include <set>

#include <fstream>

using namespace std;

int main() {

ifstream in;

ofstream out;

in.open("sample1.txt");

out.open("p2.txt");

if (in.fail() || out.fail()) {

exit(1);

}

set<string> s;

string temp;

while (!in.eof()) {

in >> temp;

s.insert(temp);

}

for (auto e:s) {

out << e << endl;

}

in.close();

out.close();

}

**Input File (sample1.txt) – The Gentle Seduction Extract**

**Output File (p2.txt) – Last 10 Lines:**

wore

work

work,

work.

worked

working

world

world,

world.

worse

worse,

would

wouldn't

wrapped

write

wrong,

wrong.

wrought

year

year.

year."

yearning

yearning,

years

years,

years.

years?

yet

yet."

yielded

you

you'd

you'll

you're

you,

you.

you."

young

young."

younger;

your

yourself,

youth

I prefer vector as it still keeps the multiples while also keeping them sorted. It's great that it's sorted because then it will take less time to find an element in the vector. Keeping multiples is also good in case you want to count how many times an element occurs in a container. A set cannot do these things, therefore, a vector is better. One other bonus is that the [] operator works so you can loop over vectors without having to use an auto loop or an iterator.

**Problem 3:**

#include <iostream>

#include <fstream>

#include <map>

#include <vector>

#include <string>

using namespace std;

int main() {

ifstream in;

in.open("sample2.txt");

if (in.fail()) {

cout << "open failed" << endl;

exit(1);

}

map<int, pair<int, vector<string> > > m;

int score;

string name;

while (!in.eof()) {

in >> score;

getline(in, name);

for (int i = 0; i <= 10; i++) {

if (score == i) {

m[i].first++;

m[i].second.push\_back(name);

}

}

}

for (int i = 0; i <= 10; i++) {

cout << i;

for (int j = 1; j <= m[i].first; j++) {

cout << "x";

}

cout << endl;

}

int input;

cout << "Input score to see names who achieved it: " << endl;

cin >> input;

for (auto e:m[input].second) {

cout << e << ", ";

}

cout << endl;

return 0;

}

**Input File (sample2.txt):**

6 Billie Jean King

5 Bonny Fisher

6 Billy Ray Cyrus

5 Miley Cyrus

6 Kim Kardashian

6 Taylor Swift

6 Adam Levine

10 Emma Watson

8 Harry Potter

5 Harry S Truman

5 Holly Golightly

10 Anna Karenina

6 Paul Walker

6 George Clooney

6 Ryan Gosling

**Output:**

0

1

2

3

4

5xxxx

6xxxxxxxxx

7

8x

9

10xx

Input score to see names who achieved it:

//User inputs 10

Emma Watson, Anna J. Karenina