**oset.cpp**

#include <unordered\_set>

#include <iostream>

#include <string>

#include <fstream>

using namespace std;

void createSet(unordered\_set<string>& spam,string& str);

// input: set,string for file input

// output: returns nothing but fills set

void checkSpam(unordered\_set<string>& spam,string& str,int& goodcount,int& spamcount);

// input: set,string for file input, 2 ints

// output: returns nothing but increments correct IP and bad IP

void rmvSpam(unordered\_set<string>& spam,string& str,int& rmvcount);

// input: set,string for file input, int

// output: returns nothing but increments remove ip

void runFileCommands();

// input: none

// output: manages program

void display(unordered\_set<string>& spam, int& gcount, int& spamcount,int& rmvcount,clock\_t& ticks);

//input: set, 4 ints

//output: none

//side effects: displays all required information

int main () {

runFileCommands();

}

//\*\*\*\* define the 4 functions whose prototypes appear above \*\*\*\*

void createSet(unordered\_set<string>& spam,string& str){

string line= "";

ifstream myfile(str);

//PART ONE IS RIGHT HERE

if(myfile.is\_open()){

while(getline(myfile,line)){

//cout<<line<<endl;

spam.insert(line);

}

}

}

// input: 1 integer value, 1 string value

// output: returns nothing but increments counter if line is blank

void checkSpam(unordered\_set<string>& spam,string& str,int& goodcount,int& spamcount){

string line = "";

spamcount = 0;

goodcount = 0;

ifstream sfile(str);

if (sfile.is\_open()){

while(getline(sfile,line)){

spamcount+=spam.count(line);

//cout<<line<<" "<<endl;

goodcount+=(1-spam.count(line));

// cout<<spamcount<<" "<<goodcount<<endl;

}

}

}

void rmvSpam(unordered\_set<string>& spam,string& str,int& rmvcount){

string line = "";

rmvcount=0;

ifstream lfile(str);

if (lfile.is\_open()){

while(getline(lfile,line)){

if (spam.count(line)==1){

spam.erase(line);

rmvcount++;

}

}

}

}

void runFileCommands(){

clock\_t ticks;

ticks = clock();

unordered\_set<string> spam;

string firstf = "spammers1.txt";

string secf = "spammers2.txt";

string check1 = "check1.txt";

string check2 = "check2.txt";

string rmv1 = "remove1.txt";

string rmv2 = "remove2.txt";

int gcount;

int bcount;

int rcount;

createSet(spam,firstf);

checkSpam(spam,check1,gcount,bcount);

rmvSpam(spam,rmv1,rcount);

display(spam,gcount,bcount,rcount,ticks);

clock\_t ticks2;

unordered\_set<string> spam2;

createSet(spam2,secf);

checkSpam(spam2,check2,gcount,bcount);

rmvSpam(spam2,rmv2,rcount);

display(spam2,gcount,bcount,rcount,ticks2);

}

void display(unordered\_set<string>& spam, int& gcount, int& spamcount,int& rmvcount,clock\_t& ticks){

cout<<"Known spammers: "<<spam.size()+rmvcount<<endl;

cout<<"Good IPs: "<<gcount<<endl;

cout<<"Bad IPs: "<<spamcount<<endl;

cout<<"Removed IPs: "<<rmvcount<<endl;

ticks = clock() - ticks;

float timer = (float)ticks/ CLOCKS\_PER\_SEC;

cout<<"Ticks of the clock: "<<timer<<endl;

}

**set.cpp**

#include <set>

#include <iostream>

#include <string>

#include <fstream>

using namespace std;

void createSet(set<string>& spam,string& str);

// input: set,string for file input

// output: returns nothing but fills set

void checkSpam(set<string>& spam,string& str,int& goodcount,int& spamcount);

// input: set,string for file input, 2 ints

// output: returns nothing but increments correct IP and bad IP

void rmvSpam(set<string>& spam,string& str,int& rmvcount);

// input: set,string for file input, int

// output: returns nothing but increments remove ip

void runFileCommands();

// input: none

// output: manages program

void display(set<string>& spam, int& gcount, int& spamcount,int& rmvcount,clock\_t& ticks);

//input: set, 4 ints

//output: none

//side effects: displays all required information

int main () {

runFileCommands();

}

//\*\*\*\* define the 4 functions whose prototypes appear above \*\*\*\*

void createSet(set<string>& spam,string& str){

string line= "";

ifstream myfile(str);

//PART ONE IS RIGHT HERE

if(myfile.is\_open()){

while(getline(myfile,line)){

//cout<<line<<endl;

spam.insert(line);

}

}

}

// input: 1 integer value, 1 string value

// output: returns nothing but increments counter if line is blank

void checkSpam(set<string>& spam,string& str,int& goodcount,int& spamcount){

string line = "";

spamcount = 0;

goodcount = 0;

ifstream sfile(str);

if (sfile.is\_open()){

while(getline(sfile,line)){

spamcount+=spam.count(line);

//cout<<line<<" "<<endl;

goodcount+=(1-spam.count(line));

// cout<<spamcount<<" "<<goodcount<<endl;

}

}

}

void rmvSpam(set<string>& spam,string& str,int& rmvcount){

string line = "";

rmvcount=0;

ifstream lfile(str);

if (lfile.is\_open()){

while(getline(lfile,line)){

if (spam.count(line)==1){

spam.erase(line);

rmvcount++;

}

}

}

}

void runFileCommands(){

clock\_t ticks;

ticks = clock();

set<string> spam;

string firstf = "spammers1.txt";

string secf = "spammers2.txt";

string check1 = "check1.txt";

string check2 = "check2.txt";

string rmv1 = "remove1.txt";

string rmv2 = "remove2.txt";

int gcount;

int bcount;

int rcount;

createSet(spam,firstf);

checkSpam(spam,check1,gcount,bcount);

rmvSpam(spam,rmv1,rcount);

display(spam,gcount,bcount,rcount,ticks);

clock\_t ticks2;

set<string> spam2;

createSet(spam2,secf);

checkSpam(spam2,check2,gcount,bcount);

rmvSpam(spam2,rmv2,rcount);

display(spam2,gcount,bcount,rcount,ticks2);

}

void display(set<string>& spam, int& gcount, int& spamcount,int& rmvcount,clock\_t& ticks){

cout<<"Known spammers: "<<spam.size()+rmvcount<<endl;

cout<<"Good IPs: "<<gcount<<endl;

cout<<"Bad IPs: "<<spamcount<<endl;

cout<<"Removed IPs: "<<rmvcount<<endl;

ticks = clock() - ticks;

float timer = (float)ticks/ CLOCKS\_PER\_SEC;

cout<<"Ticks of the clock: "<<timer<<endl;

}