|  |  |
| --- | --- |
| **Author Identification Block** | |
| **Author:** | Chris Graff |
| **Student ID:** | \*20274911 |
| **E-Mail:** | [cgraff@uco.edu](mailto:cgraff@uco.edu) |
| **Course:** | CMSC 2613 – Programming 2 |
| **CRN:** | 21641, Spring 2012 |
| **Project:** | p07 |
| **Due:** | March 16, 2012 |
| **Account:** | tt025 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Scoring Block** | | | |
| **Component** | **Available** | **Earned** | **Explanation** |
| Compilation |  |  |  |
| Submission Instructions | 2 |  |  |
| Author Identification | 1 |  |  |
| Modularity | 3 |  |  |
| Command Line | 3 |  |  |
| Input file | 3 |  |  |
| Output file | 3 |  |  |
| Execution | 10 |  |  |
| **Total** | **25** |  |  |

#include <iostream>

#include <fstream>

#include <iomanip>

#include <limits.h>

#include <limits>

#include <cstring>

#include <string>

#include "Set07.h"

#include "List07.h"

//----------------------------------------------------------------

//Author: Chris Graff

//StudentID#: \*20274911

//Email: cgraff@uco.edu

//Course: CMSC2613 Programming II

//CRN: 21641, Spring 2012

//Project: p07

//Due: March 16th, 2012

//Account: tt025

//----------------------------------------------------------------

using namespace std;

struct CommandLineException

{

CommandLineException (int max, int actual)

{

cout <<endl;

cout <<"Too many command line arguements.";

cout <<endl;

cout <<"A maximum of " <<max <<" arguements are permitted.";

cout <<endl;

cout <<actual <<" arguements were entered.";

cout <<endl;

}

};

struct FileException

{

FileException (const char\* filename)

{

cout <<endl;

cout <<"File " <<filename <<" could not be opened or doesn't exist";

cout <<endl;

}

};

void SetManager (ifstream& i1, ifstream& i2, ifstream& i3, ostream& o)

{

Set<int> s1(INT\_MIN, i1, 100);

s1.Print("set 1",o);

Set<int> s2(INT\_MIN, i2, 100);

s2.Print("set 2",o);

Set<int> s3(INT\_MIN, i3, 100);

s3.Print("set 3",o);

Set<int> I(INT\_MIN, 100);

I.Intersection(s1,s2);

I.Print("set I",o);

Set<int> U(INT\_MIN, 100);

U.Union(s2,s3);

U.Print("set U",o);

Set<int> D(INT\_MIN, 100);

D.Difference(U,I);

D.Print("set D",o);

}

int main (int argc,const char\* argv[])

{

char ifn1[255],ifn2[255], ifn3[255], ofn[255];

switch (argc)

{

case 1:

cout << "Enter the input file name 1:";

cin >> ifn1;

cout << "Enter the input file name 2:";

cin >> ifn2;

cout << "Enter the input file name 3:";

cin >> ifn3;

cout << "Enter the output file name:";

cin >> ofn;

break;

case 2:

strcpy(ifn1,argv[1]);

cout << "Enter the input file name 2:";

cin >> ifn2;

cout << "Enter the input file name 3:";

cin >> ifn3;

cout << "Enter the output file name:";

cin >> ofn;

break;

case 3:

strcpy(ifn1,argv[1]);

strcpy(ifn2,argv[2]);

cout << "Enter the input file name 3:";

cin >> ifn3;

cout << "Enter the output file name:";

cin >> ofn;

break;

case 4:

strcpy(ifn1,argv[1]);

strcpy(ifn2,argv[2]);

strcpy(ifn3,argv[3]);

cout << "Enter the output file name:";

cin >> ofn;

break;

case 5:

strcpy(ifn1,argv[1]);

strcpy(ifn2,argv[2]);

strcpy(ifn3,argv[3]);

strcpy(ofn,argv[4]);

break;

default:

throw CommandLineException (4, argc-1);

break;

}

ifstream i1 (ifn1); if(!i1) throw FileException(ifn1);

ifstream i2 (ifn2); if(!i2) throw FileException(ifn2);

ifstream i3 (ifn3); if(!i3) throw FileException(ifn3);

ofstream o (ofn); if(!o) throw FileException(ofn);

SetManager(i1, i2, i3, o);

i1.close();

i2.close();

i3.close();

o.close();

return 0;

}

#ifndef List07\_h

#define List07\_h

#include <iostream>

#include <iomanip>

#include <fstream>

#include <string>

#include <limits>

//----------------------------------------------------------------

//Author: Chris Graff

//StudentID#: \*20274911

//Email: cgraff@uco.edu

//Course: CMSC2613 Programming II

//CRN: 21641, Spring 2012

//Project: p07

//Due: March 16th, 2012

//Account: tt025

//----------------------------------------------------------------

using namespace std;

struct ListException

{

ListException (string m)

{

cout << endl;

cout << "The List is " << m << ".";

cout << endl;

}

};

template <class T>

class List

{

int size;

int count;

int cursor;

T\* L;

void New (int sz) {size = sz; count = 0; cursor = 0; L = new T[size]; L[0] = MIN;}

int Index (T key)

{

int lo = 1;

int hi = count;

while (lo <= hi)

{

int m = (lo + hi) / 2;

if (key == L[m]) return m;

if (key < L[m]) hi = m - 1; else lo = m + 1;

}

return 0;

}

protected:

const T MIN;

public:

void Empty(void)

{

int sz = size;

if (L) delete[] L;

New (sz);

}

List(T m, int sz = 100):size(sz), count(0), cursor(0), MIN(m)

{

L = new T[size];

L[0] = MIN;

}

List(T m, istream& i, int sz = 100):size(sz), count(0), cursor(0), MIN(m)

{

L = new T[size];

L[0] = MIN;

Scan(i);

}

~List(){if (L) delete[] L;}

bool IsEmpty(void){return count <= 0;}

bool IsFull(void){return count > size - 1;}

void Insert(T key)

{

if (IsMember(key)) return;

if (IsFull()) throw ListException("full");

int i =++count;

for ( ; key < L[i - 1]; i--) L[i] = L[i - 1];

L[i] = key;

}

void Remove(T key)

{

int i = Index(key);

if (i == 0) return;

for ( ; i < count; i++)L[i] = L[i +1];

count--;

}

void Print(string title, ostream& o)

{

o <<title <<"{";

for (int a = 1; a <= count; a++)

{

if (a>1) o <<",";

o <<L[a];

}

o <<"}" <<endl;

}

void Scan(istream& i)

{

for ( ; ; )

{

T key;

i >> key;

if (i.eof()) break;

Insert(key);

}

}

void First (void)

{

cursor = 1;

}

void Next (void)

{

if (cursor < count + 1) cursor++;

}

bool IsEol (void)

{

return cursor > count;

}

T Key(void)

{

if (cursor) return L[cursor];

}

bool IsMember (T key)

{

return Index (key);

}

};

#endif

#ifndef Set07\_h

#define Set07\_h

#include <iostream>

#include <iomanip>

#include <fstream>

#include <string>

#include <limits>

#include "List07.h"

//----------------------------------------------------------------

//Author: Chris Graff

//StudentID#: \*20274911

//Email: cgraff@uco.edu

//Course: CMSC2613 Programming II

//CRN: 21641, Spring 2012

//Project: p07

//Due: March 16th, 2012

//Account: tt025

//----------------------------------------------------------------

using namespace std;

template <class T>

class Set: public List<T>

{

public:

Set (T m, int sz = 100):List<T> (m,sz){}

Set (T m, istream& i, int sz = 100):List<T> (m,i,sz){}

~Set (){}

void Intersection (Set& s1,Set& s2)

{

List <T>::Empty ();

for (s1.First (); !s1.IsEol (); s1.Next ())

{

if (s2.IsMember(s1.Key ())) Insert(s1.Key ());

}

}

void Union (Set& s1, Set& s2)

{

List <T>::Empty ();

for (s1.First ();!s1.IsEol ();s1.Next ())

List <T>::Insert (s1.Key ());

for (s2.First ();!s2.IsEol ();s2.Next ())

List <T>::Insert (s2.Key ());

}

void Difference (Set& M,Set& S)

{

List<T>::Empty ();

for (M.First (); !M.IsEol (); M.Next ())

{

Insert(M.Key());

}

for (List <T>::First (); !List<T>::IsEol (); List <T>::Next ())

{

if (S.IsMember (List <T>::Key())) Remove (List <T>::Key());

}

}

};

#endif