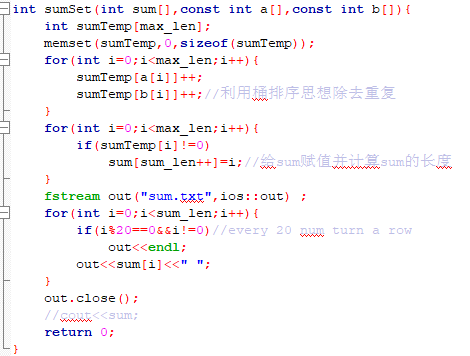
# The second report—曹鹏霄

**development environment：Devc++**

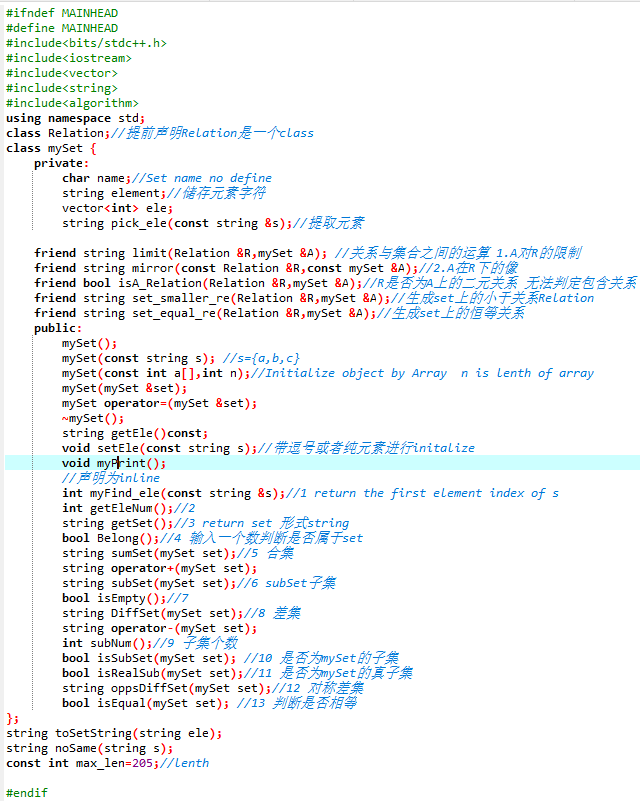
In the first week, I tried to take a way how to make up my codes programming ago. We are supposed to make a class to Integrate these pieces of codes.

I found most of previous codes are allowed to use in the class, I only need change their parameters, return value or type.

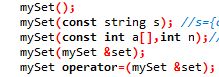
such as, the code above, I just need change its return type into class type defined by myself, then change its parameters into empty parameter and delete operator of file. Hence, I get finished one member function. Others could do like this as well.

Then, it’s my Programming process.

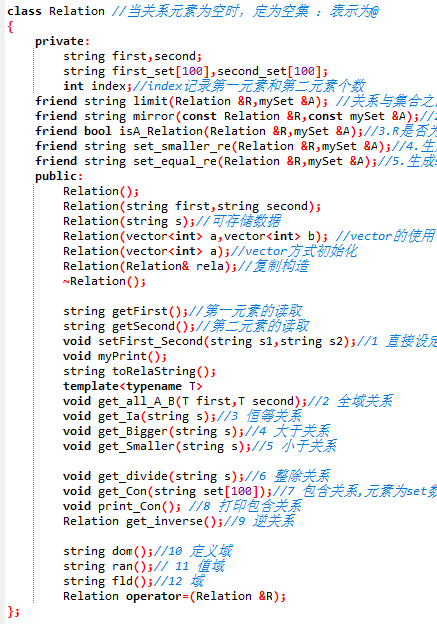
First, I created a class named mySet and it has two data in private data field, object name and element set elements. They are both of string type. Show in following picture.



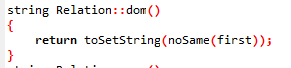
Then, I opened previous cpp documents, replace their parameter type with mySet. And replace the return type with string which is their element string.

About what I had finish a set class it has 4 constructors a copy constructor and a destructor but the destructor has not any operation now, also 10 set operations and I use some const and private to make sure some security of data, vector is used as well.

Two weeks later, I almost finished the assign, the implementation of the copy constructor, at least 4 constructors in each class, implementation of the destructor, apply vector class, data is stored in files, member variables in class to store the set and relation and they are stored in private data field, inline function is also used in mySet class(string pick\_ele(const string s)).

In second week, I mainly completed the implementation of part Relation class and operator between Sets and Relations.

In this project, I find it could be used with many common algorithms. So I write these same parts as a function so that I can invoke them directly next time.



Problem: When using multiple files, you will encount many problem you never encounted in past, such as Multiple inclusion to make codes compile fail.

How to complete the requirement

There are some of codes and They are distributed in different files :

Class:

Set:

*class mySet {*

*private:*

*char name;//Set name no define*

*string element;//储存元素字符*

*vector<int> ele;*

*string pick\_ele(const string &s);*

*public:*

*mySet();*

*mySet(const string s); //s={a,b,c}*

*mySet(const int a[],int n);//Initialize object by Array n is lenth of array*

*mySet(mySet &set);*

*~mySet();*

*string getEle();*

*void setEle(const string s);//带逗号或者纯元素进行initalize*

*void myPrint();*

*int myFind\_ele(const string &s);//return the first element index of s*

*int getEleNum();*

*bool Belong();//输入一个数判断是否属于set*

*string sumSet(mySet set);//合集*

*string subSet(mySet set);//subSet子集*

*bool isEmpty();*

*string DiffSet(mySet set);//差集*

*int subNum();//子集个数*

*bool isSubSet(mySet set); //是否为mySet的子集*

*bool isRealSub(mySet set);//是否为mySet的真子集*

*string oppsDiffSet(mySet set);//对称差集*

*bool isEqual(mySet set); //判断是否相等*

*};*

## Relation

#ifndef RELATION

#define RELATION

#include"mainHead.h"

class Relation //当关系元素为空时，定为空集 ：表示为@

{

private:

string first,second;

string first\_set[100],second\_set[100];

int index;//index记录第一元素和第二元素个数

friend string limit(Relation &R,mySet &A); //关系与集合之间的运算 1.A对R的限制

friend string mirror(const Relation &R,const mySet &A);//2.A在R下的像

friend bool isA\_Relation(Relation &R,mySet &A);//3.R是否为A上的二元关系

friend string set\_smaller\_re(Relation &R,mySet &A);//4.生成set上的小于关系Relation

friend string set\_equal\_re(Relation &R,mySet &A);//5.生成set上的恒等关系

public:

Relation();

Relation(string first,string second);

Relation(string s);//可存储数据

Relation(vector<int> a,vector<int> b); //vector的使用

Relation(vector<int> a);//vector方式初始化

Relation(Relation& rela);//复制构造

~Relation();

string getFirst();//第一元素的读取

string getSecond();//第二元素的读取

void setFirst\_Second(string s1,string s2);//1 直接设定第一第二元素

void myPrint();

string toRelaString();

template<typename T>

void get\_all\_A\_B(T first,T second);//2 全域关系

void get\_Ia(string s);//3 恒等关系

void get\_Bigger(string s);//4 大于关系

void get\_Smaller(string s);//5 小于关系

void get\_divide(string s);//6 整除关系

void get\_Con(string set[100]);//7 包含关系,元素为set数组

void print\_Con(); //8 打印包含关系

Relation get\_inverse();//9 逆关系

string dom();//10 定义域

string ran();// 11 值域

string fld();//12 域

Relation operator=(Relation &R);

};

#endif

## 运行图：

