## 二元关系的实现：完成了11个功能

1. 计算从A到B的二元关系的结果
2. A上的二元关系的全集
3. 恒等集
4. 小于或等于
5. 大于
6. 整除关系
7. 包含关系
8. 逆关系
9. 定义域
10. 值域
11. 域

其中“@”代表空集

##### 代码：

###### MainHead.h

#ifndef MAINHEAD

#define MAINHEAD

#include<bits/stdc++.h>

using namespace std;

class openFile{//打开文件的class

private:

public:

fstream io;

openFile(string filename){

io.open(filename.c\_str(),ios::in|ios::app);

}

~openFile(){

io.close();

}

};

string binRela(string s1,string s2);

string binRela(string s1,string s2,bool flag);

string getA\_B();//计算从A到B的二元关系的结果

string getA\_A(); //A上的二元关系的全集

string get\_Ia();//恒等集

string get\_La();//小于或等于

template <typename T>

bool La\_flag(T a,T b);

string get\_BigerA();//大于

template <typename T>

bool BigerA\_flag(T a,T b);

string get\_Db();//整除关系

bool Db\_flag(int a,int b);

string get\_Ra();//包含关系

int mySub(string,string\*);

string get\_inverse\_R();//逆关系

int myFind(string s);

void myNoSame(string&);

string domR();//定义域

string ranR();//值域

string fldR();//域

#endif

###### Main.cpp

#include "mainHead.h"

int main()

{

getA\_B();

getA\_A();

get\_Ia();

get\_La();

get\_BigerA();

get\_Db();//整除关系

get\_Ra();

get\_inverse\_R();//逆关系

domR();

ranR();

fldR();

return 0;

}

###### getA\_B.cpp

#include"mainHead.h"

string getA\_B(){

openFile A\_B("binRela.txt");//openFile为class

string s;

A\_B.io>>s;

cout<<s<<endl;

string A,B;//A 为小写字母B为数字

for(int i=0; i<s.size(); i++) {

if(s[i]>='0'&&s[i]<='9')

B+=s[i];

else if(s[i]>='a'&&s[i]<='z')

A+=s[i];

}

cout<<A<<B<<endl;

string ans=binRela(A,B);

cout<<"AxB="<<ans<<endl;

A\_B.io<<"AxB="<<ans<<endl;//写入文件

return ans;

}

###### getA\_A.cpp

#include"mainHead.h"

string getA\_A(){

openFile A\_A("binRela.txt");//openFile为class

string s;

A\_A.io>>s;

// cout<<s<<endl;

string A;//A 为小写字母

for(int i=0; i<s.size(); i++)

if(s[i]>='a'&&s[i]<='z')

A+=s[i];

string ans=binRela(A,A);

A\_A.io<<"AXA="<<ans<<endl;

cout<<"AXA="<<ans<<endl;

return ans;

}

###### subFun.cpp

#include"mainHead.h"

string binRela(string A,string B){//生成AxB

string ans("{}");

for(int i=0;i<A.size();i++){

for(int j=0;j<B.size();j++){

stringstream ss;

ss<<"<"<<A[i]<<","<<B[j]<<">";

if(i!=A.size()-1||j!=B.size()-1)

ss<<",";

//cout<<ss.str()<<" "<<5+(i\*j+j)\*6<<endl;

ans.insert(1+(i\*B.size()+j)\*6,ss.str());

}

}//cout<<ans<<endl;

return ans;

}

string binRela(string A,string B,bool flag){//生成AxB A和B满足关系flag

string ans("{}");

int pos=0;//用于 记录位置

for(int i=0;i<A.size();i++){

for(int j=0;j<B.size();j++){

if(flag==true){

stringstream ss;

ss<<"<"<<A[i]<<","<<B[j]<<">"<<",";

//cout<<ss.str()<<" "<<1+(i\*j+j)\*6<<endl;

ans.insert(1+pos\*6,ss.str());

pos++;

}

}

}

ans.erase(ans.size()-2,1);

cout<<ans<<endl;

return ans;

}

int myFind(string s){//寻找数字字符并返回位置

for(char i='0';i<='9';i++){

int pos=s.find(i);

if(pos!=s.npos)

return pos;

}

}

void myNoSame(string& s){

int a[300];

memset(a,0,sizeof(a));

for(int i=0;i<s.size();i++){

a[s[i]]++;

}

s.clear();

for(int i=0;i<300;i++)

if(a[i]!=0)

s+=i;

}

###### Get\_Ia.cpp

#include"mainHead.h"

string get\_Ia(){

openFile Ia("binRela.txt");//openFile为class

string s;

Ia.io>>s;

//cout<<s<<endl;

string A;//A 为小写字母

for(int i=0; i<s.size(); i++)

if(s[i]>='a'&&s[i]<='z')

A+=s[i];

// cout<<A<<endl;//

string ans("{}");

for(int i=0;i<A.size();i++){

stringstream ss;

ss<<"<"<<A[i]<<","<<A[i]<<">";

if(i!=A.size()-1)

ss<<",";

//cout<<ss.str()<<" "<<5+(i\*j+j)\*6<<endl;

ans.insert(1+i\*6,ss.str());

}

cout<<"Ia="<<ans<<endl;

Ia.io<<"Ia="<<ans<<endl;

return ans;

}

###### Get\_La.cpp

#include"mainHead.h"

template <typename T>

bool La\_flag(T a,T b){

return a<=b?true:false;

}

string get\_La(){

openFile La("binRela.txt");//openFile为class

string s;

La.io>>s;

//cout<<s<<endl;

string A;//A 为小写字母

for(int i=0; i<s.size(); i++)

if(s[i]>='a'&&s[i]<='z')

A+=s[i];

string ans("{}");

int pos=0;//用于 记录位置

for(int i=0;i<A.size();i++){

for(int j=0;j<A.size();j++){

if(La\_flag(A[i],A[j])==true){

stringstream ss;

ss<<"<"<<A[i]<<","<<A[j]<<">"<<",";

//cout<<ss.str()<<" "<<1+(i\*j+j)\*6<<endl;

ans.insert(1+pos\*6,ss.str());

pos++;

}

}

}

ans.erase(ans.size()-2,1);

cout<<"La="<<ans<<endl;//小于或等于关系

La.io<<"La="<<ans<<endl;

return ans;

}

###### Get\_Db.cpp

#include"mainHead.h"

bool Db\_flag(int a,int b){//a整除b,a是b的因子

if(a==0||b==0)return false;

return b%a==0?true:false;

}

string get\_Db(){

openFile Db("binRela.txt");//openFile为class

string s;

Db.io>>s;

//cout<<s<<endl;

string B;//B为数字

for(int i=0; i<s.size(); i++)

if(s[i]>='0'&&s[i]<='9')

B+=s[i];

string ans("{}");

int pos=0;//用于 记录位置

for(int i=0;i<B.size();i++){

for(int j=0;j<B.size();j++){

if(Db\_flag(B[i]-'0',B[j]-'0')==true){

stringstream ss;

ss<<"<"<<B[i]<<","<<B[j]<<">"<<",";

//cout<<ss.str()<<" "<<1+(i\*j+j)\*6<<endl;

ans.insert(1+pos\*6,ss.str());

pos++;

}

}

}

ans.erase(ans.size()-2,1);

cout<<"Db="<<ans<<endl;//小于或等于关系

Db.io<<"Db="<<ans<<endl;

return ans;

}

###### getBiggerA.cpp

#include"mainHead.h"

template <typename T>

bool BigerA\_flag(T a,T b){

return a>b?true:false;

}

string get\_BigerA(){

openFile BigerA("binRela.txt");//openFile为class

string s;

BigerA.io>>s;

// cout<<s<<endl;

string A;//A 为小写字母

for(int i=0; i<s.size(); i++)

if(s[i]>='a'&&s[i]<='z')

A+=s[i];

string ans("{}");//cout<<A<<endl;

int pos=0;//用于 记录位置

for(int i=0;i<A.size();i++){

for(int j=0;j<A.size();j++){

if(BigerA\_flag(A[i],A[j])==true){

stringstream ss;

ss<<"<"<<A[i]<<","<<A[j]<<">"<<",";

//cout<<ss.str()<<" "<<1+(i\*j+j)\*6<<endl;

ans.insert(1+pos\*6,ss.str());

pos++;

}

}

}

ans.erase(ans.size()-2,1);

cout<<"BigerA="<<ans<<endl;//大于关系

BigerA.io<<"BigerA="<<ans<<endl;

return ans;

}

###### Get\_Ra.cpp

#include"mainHead.h"//把定义为空集

int mySub(string s,string subS[]){

int pos=s.find('{')+1;

int num=0;//int test=0;

for(int i=pos;i<s.size();){

if(s[i]=='{'){

while(s[i+1]!='}'){//cout<<"test:"<<test++<<endl;

if(i+1>s.size())break;

subS[num]+=s[i+1];

i++;

}

num++;//第i个子集

i++;

}

else if(s[i]=='@'){//检测空集

subS[num]="@";

num++;

i++;//cout<<s[i];

}

else {//cout<<s[i];

i++;

}

}

//cout<<"num="<<num<<endl;

return num;

}

bool Ra\_flag(string a,string b){//b包含a的判断条件

if(a=="@")return true;

return b.find(a)!=b.npos?true:false;

}

string get\_Ra(){

openFile Ra("get\_Ra.txt");

string s,\*subS=new string[100];

Ra.io>>s;

//cout<<s<<endl;

int len=mySub(s,subS); //元素个数

for(int i=0;i<len;i++){//输出提取到的集合元素

//cout<<subS[i]<<endl;

}

string ans("{");//答案

int pos=0;//用于 记录位置

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

for(int j=0;j<len;j++){

if(Ra\_flag(subS[i],subS[j])==true){

stringstream ss;

ss<<"<"<<"{"<<subS[i]<<"}"<<","<<"{"<<subS[j]<<"}"<<">"<<",";

//cout<<ss.str()<<" "<<endl;

ans+=ss.str();

pos++;

}

}

}

ans.append("}");

ans.erase(ans.size()-2,1);

cout<<"Ra="<<ans<<endl;//小于或等于关系

Ra.io<<"Ra="<<ans<<endl;

return s;

}

###### **Get\_inverse\_R.cp**p

#include"mainHead.h"

string get\_inverse\_R(){

openFile inverse\_R("R.txt");

string s;

inverse\_R.io>>s;

cout<<s<<endl;

string ans=s;//建立答案

int pos=myFind(ans);

while(pos<ans.size()){

swap(ans[pos],ans[pos+2]);

pos+=6;

}

ans.erase(0,2);

cout<<"R^-1="<<ans<<endl;

inverse\_R.io<<"R^-1="<<ans<<endl;

return ans;

}

###### domR.cpp

#include"mainHead.h"

string domR(){

openFile domR("R.txt");

string s;

domR.io>>s;

//cout<<s<<endl;

int pos=myFind(s);//cout<<pos<<":"<<s[pos]<<endl;

string ans("{");//cout<<"invoked"<<endl;

stringstream ss;

while(pos<s.size()){

ss<<s[pos]<<",";

ans+=ss.str();

ss.str("");//清空ss的缓存

pos+=6;

}

ans.replace(ans.size()-1,1,"}");

cout<<"domR="<<ans<<endl;

domR.io<<"domR"<<ans<<endl;

return ans;

}

string ranR(){

openFile ranR("R.txt");

string s;

ranR.io>>s;

//cout<<s<<endl;

int pos=myFind(s)+2;//cout<<pos<<":"<<s[pos]<<endl;

string ans("{");//cout<<"invoked"<<endl;

stringstream ss;

while(pos<s.size()){

ss<<s[pos]<<",";

ans+=ss.str();

ss.str("");//清空ss的缓存

pos+=6;

}

ans.replace(ans.size()-1,1,"}");

cout<<"ranR="<<ans<<endl;

ranR.io<<"ranR"<<ans<<endl;

return ans;

}

string fldR(){

openFile fldR("R.txt");

string s;

fldR.io>>s;

string ansTemp;

for(int i=0;i<s.size();i++)

if(s[i]>='0'&&s[i]<='9')

ansTemp+=s[i];

myNoSame(ansTemp);

//cout<<ansTemp<<endl;

string ans("{");

stringstream ss;

for(int i=0;i<ansTemp.size();i++){

ss<<ansTemp[i]<<",";

ans+=ss.str();

ss.str("");//清空ss的缓存

}

ans.replace(ans.size()-1,1,"}");

cout<<"fldR="<<ans<<endl;

fldR.io<<"fldR="<<ans<<endl;

return ans;

}

##### C:\Users\shinelon\AppData\Roaming\Tencent\Users\2909371614\TIM\WinTemp\RichOle\{)KFXJ[K75Q52N0)H(5TTUR.png运行图：

**“@”代表空集**

