**宁夏理工学院**

实验报告



课程名称： 离散数学

姓 名： 陈毅

班 级： 计类20104班

学 号： 45420323

1. **实验名称**

实验五：集合运算

**二、实验目的**

掌握用计算机求集合的交、并、差和补运算的方法。

1. **实验内容**

编程实现集合的交、并、差和补运算。

**四、实验环境**

VC6.0/Dev c++

1. **实验原理和方法**

（1）用数组A，B，C，E表示集合。输入数组A，B，E（全集），输入数据时要求检查数据是否重复（集合中的数据要求不重复），要求集合A，B是集合E的子集。

以下每一个运算都要求先将集合C置成空集。

（2）二个集合的交运算：把数组A中元素逐一与数组B中的元素进行比较，将相同的元素放在数组C中，数组C便是集合A和集合B的交。

（3）二个集合的并运算：把数组A中各个元素先保存在数组C中。将数组B中的元素逐一与数组B中的元素进行比较，把不相同的元素添加到数组C中，数组C便是集合A和集合B的并。

（4）二个集合的差运算：把数组A中各个元素先保存在数组C中。将数组B中的元素逐一与数组B中的元素进行比较，把相同的元素从数组C中删除，数组C便是集合A和集合B的差A-B。

（5）集合的补运算：将数组E中的元素逐一与数组A中的元素进行比较，把不相同的元素保存到数组C中，数组C便是集合A关于集合E的补集。

求补集是一种种特殊的集合差运算。

**六、实验步骤及实现过程**

1.设计思路

（1）首先通过主函数choice自己想执行的运算，通过switch来控制五个基本关系运算。

（2）通过构造函数分别求出各个集合运算，利用循环和条件来控制每个元素依次执行遍历。

1. 具体程序

#include<stdio.h>

#include<stdlib.h>

int a[50]={0},b[50]={0},c[50]={0},d[50]={0},e[50]={0};

//显示输出

void show (){

printf("1---输入集合A和集合B\n");

printf("2---求集合A交B\n");

printf("3---求集合A并B\n");

printf("4---求集合A-B\n");

printf("退出，输入一个负数！");

}

//输入集合A和B

void input(int x,int y){

int i,j;

printf("请输入A的元素");

for(i=0;i<x;i++){

scanf("%d",&a[i]);

getchar();

for(j=0;j<i;j++){ //通过循环判断进行查重

if(a[i]==a[j]){

printf("输入错误！请重新输入");

a[i]=0;

i-- ;

}

}

}

printf("请输入B的元素");

for(i=0;i<y;i++){

scanf("%d",&b[i]);

getchar();

for(j=0;j<i;j++){ //通过循环判断进行查重

if(b[i]==b[j]){

printf("输入错误！请重新输入");

b[i]=0;

}

}

}

}

//求集合A交B，因为交集会多次使用，故与输出分开。

int hand(int x,int y){

int i,j,k=0;

for(i=0;i<x;i++){

for(j=0;j<y;j++){

if(a[i]==b[j]){

c[k]=a[i];

k++;

}

}

}

return k;

}

//输出交集

void outhand(int k){

int i;

printf("A交B={");

for(i=0;i<k-1;i++){

printf("\t %d,",c[i]);

}

printf("\t %d}\n",c[k-1]);

}

//求集合A并B

void combine(int x,int y,int k){

int i,j,m=0,n;

int com[50]={0};

for(i=0;i<x;i++){

d[m]=a[i];

m++;

}

for(i=0;i<y;i++){

d[m]=b[i];

m++; //m为a和b集合的总的元素个数

}

/\*

for(i=0;i<x+y-k;i++){ //x+y-k为并集的实际元素个数 ,且该循环除去重复元素

for(j=m;j>i;j--){

if(d[j]==d[i]){

d[j]=d[j+1]; //用重复元素后一位元素覆盖该元素

i--; //从新进行该轮循环，消除上一步带来的多个元素重复的影响

break; //跳出里层for循环

}

}

}

\*/

for(i=0;i<x+y-k;i++){ //x+y-k为并集的实际元素个数 ,且该循环除去重复元素

for(j=i+1;j<m;j++){

if(d[j]==d[i]){

for(n=j;n<m;n++){

d[n]=d[n+1];

}

break;

}

}

}

printf("A并B={");

for(i=0;i<x+y-k-1;i++){

printf("\t %d,",d[i]);

}

printf("\t %d}\n",d[x+y-k-1]);

}

//求集合A-B

void reduce(int x,int k){

int i,j,m;

for(i=0;i<x;i++){

e[i]=a[i];

}

for(i=0;i<x-k;i++){ //思路同并集中除去重复元素

for(j=0;j<k;j++){

if(e[i]==c[j]){

for (m=i;m<x;m++){ //将重复后所有元素前移

e[m]=e[m+1];

}

i--;

break;

}

}

}

printf("A-B={");

for(i=0;i<x-k-1;i++){

printf("\t %d,",e[i]);

}

printf("\t %d}\n",e[x-k-1]);

}

int main(){

int x,y,i,j,k,choice;

while(1){

show();

printf("请选择你需要进行的操作\n");

scanf("%d",&choice);

if(choice<0){

printf("已退出。");

return 0;

}else{

switch(choice){

case 1:

printf("请输入A集合中元素个数\n");

scanf("%d",&x);

printf("请输入B集合中元素个数\n");

scanf("%d",&y);

input(x,y);

k=hand(x,y);

printf("请按回车继续！\n");

getchar();

system("cls");

break;

case 2:

outhand(k);

printf("请按回车继续！\n");

getchar();

getchar();

system("cls");

break;

case 3:

combine(x,y,k);

printf("请按回车继续！\n");

getchar();

getchar();

system("cls");

break;

case 4:

reduce(x,k);

printf("请按回车继续！\n");

getchar();

getchar();

system("cls");

break;

default :

printf("输入有误！\n");

printf("请按回车继续！\n");

getchar();

getchar();

system("cls");

break;

}

}

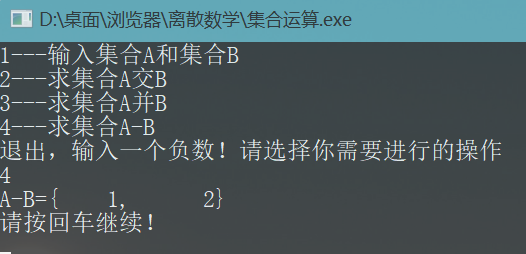
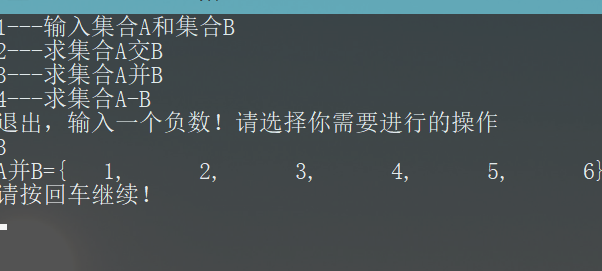
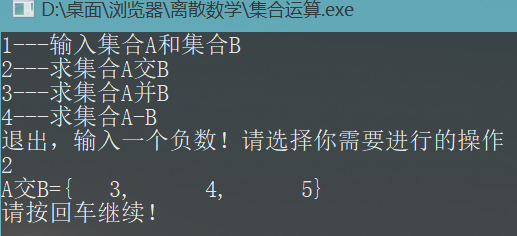
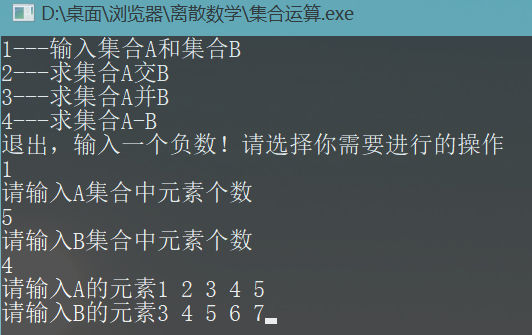
}

return 0;

}

**七、实验输入输出的数据**

**（类似如下截屏）**

****

1. **心得体会**

**该实验我没有用到链表来进行运算，而只是简单的采用构造函数的方法来计算各个集合，通过程序遍历数组的各个元素来实现是否满足关系运算，也在程序中实现了运算界面，可以通过自己选择要计算哪一个运算，使程序更加规划化。**