实验一　线性表

班级:计算机 17-1 班_姓名:__冯子旋_学号：__07__

一、实验目的

1．掌握线性表的一种实现方法；

2．掌握利用线性表解决实际问题的方法；

3．通过上机实践加强利用数据结构解决实际应用应用问题的能力。

二、实验题目与要求

实验题目：顺序表的实现及其应用

实验要求：

（1）实现顺序表的类型定义及顺序表的初始化、插入、删除、取元素、输出等操作。

（2）以采用顺序表表示集合，设计程序实现集合的创建、并集及交集等运算。

三、实验内容

1．顺序表的实现

完成顺序表的类型定义及其基本操作函数代码，并将其保存为“SqList.h”。

```
//顺序表类型及其基本操作函数的定义
#define ListSpaceIncr 20
typedef struct{
    LElemType *base;
    int length;
    int listSize;
}SqList; //SqList 为顺序表类型
Status InitSqList(SqList &L, int InitSize){
//创建初始空间大小为 InitSize 的空顺序表
  L.base = (LElemType * )malloc(InitSize * sizeof(LElemType));
   if(!L.base)
        return(OVERFLOW);
   L.length = 0;
   L.listSize = InitSize;
   return OK;
}
int listLength(SqList L){//求顺序表 L 的长度
   return L.length;
}
Status listIsEmpty(SqList L) {
//判断顺序表 L 是否为空；空返回 TRUE
//否则返回 FALSE
  if(!L.length)
        return TRUE;
   else return FALSE;
}
void clearList(SqList &L){
//将顺序表 L 清空
   L.length = 0;
}
Status getElem(SqList L,int i,LElemType &e){
//取顺序表 L 的第 i 个元素
   if(!L.length)    return ERROR;
   e = L.base[i-1];
   return OK;
}
Status equal(LElemType, LElemType);
        //元素相等关系的比较函数声明
int locateElem(SqList L, LElemType e){
//在顺序表 L 中查找与 e 相等的第一个元素
//并返回其位序，查找失败时返回 0
  int i = 0;
   while (i<L.length && !equal(L.base[i],e))
        i++;
   if(i<L.length)  return i+1;
   else return 0;
}
Status listInsert(SqList &L, int i, LElemType e){
//在顺序表 L 的第 i 个位置插入元素 e
  LElemType *newBase;
   int j;
   if(i<1 || i>L.length+1)    return ERROR;
   if(L.length==L.listSize)
   {
        newBase =
```

```
(LElemType*)realloc(L.base,(L.listSize+ListSpaceIncr)*sizeof(LElemType));
        if(!newBase)   return OVERFLOW;
        L.base = newBase;
        L.listSize+=ListSpaceIncr;
    }
    for(j=L.length-1; j>=i-1; j--)
    {
        L.base[j+1] = L.base[j];
    }
    L.base[i-1] = e;
    L.length++;
    return OK;
}
Status listDelete(SqList &L,int i,LElemType &e){
    //删除顺序表的第 i 个元素并由 e 返回其值
   int j;
    if(i<1 || i>L.length) return ERROR;
    e = L.base[i-1];
    for(j=i; j<L.length; j++)
    {
        L.base[j-1] = L.base[j];
    }
    L.length--;
    return OK;
}
void listTraverse(SqList L,void (*visit)(LElemType e))
{   //遍历操作
    int i;
    for(i=0;i<L.length;i++)
        visit(L.base[i]);
}
```

2．顺序表应用程序设计

以采用顺序表表示集合，设计程序实现集合的创建、并集及交集等运算。

```
#include <stdio.h>
#include <stdlib.h>
#include "Status.h"
typedef int LElemType; //定义元素类型为 int 类型
#include "SqList.h"
typedef SqList mySetType; //定义集合类型
Status equal(LElemType e1,LElemType e2){
    //元素相等关系的比较函数
    if(e1==e2) return TRUE;
    else return FALSE;
}
void visitElem(LElemType e){//元素访问操作
    printf("%d ",e);
}
void creatSet(mySetType &A, int n){
//通过输入的 n 个元素创建集合 A
    int i,e;
    InitSqList(A, n+10);
    printf("\t 输入%d 个整数：",n);
    for (i=0; i<n; i++){
        scanf("%d", &e);
        listInsert(A,i+1,e) ;    //尾端插入元素
    }
}
void setUnion(mySetType A,mySetType B,mySetType &C)
{//集合的并集运算，实现 C=A∪B
    int i,k,len,e;
    clearList(C); k=0; // C 清空；用 k 保存 C 的长度
    len=listLength(A);
    for(i=1; i<=len; i++){
    //将集合 A 中所有元素在集合 C 的尾端插入
        getElem(A,i,e);
        listInsert(C, ++k , e);
    }
    len=  listLength(B) ;      //求集合 B 的长度
    for(i=1;i<=len;i++){
        //将 B 中不属于 A 的元素在 C 尾端插入
        getElem(B,i,e); //取 B 的第 i 个元素，赋给 e
        if(!locateElem(A, e)) //若元素 e 不在 A 中
            listInsert(C, ++k, e); //在 C 的尾端插入 e
    }
}
void setIntersection(mySetType &A,mySetType B) {//集
合交集运算，实现 A=A∩B
    int i,e,len;
    len=listLength(A); i=1;
```

```
    while(i<=len){
        getElem(A, i, e);       //取 A 的第 i 个元素给 e
        if(  !locateElem(B,e) ) { //若 e 不属于 B
           listDelete(A,i,e) ;    //删除 A 的第 i 个元素
            len--; //当前 A 的长度已减 1；
                  //下一元素位序仍为 i
        }
        else i++;下一要处理元素的位序为 i+1
    }
}

void outputSet(mySetType A){//集合的输出操作
    printf("{ ");
    listTraverse(A,visitElem);
    printf("}\n");
}

int main(){ //用于测试的主函数
      mySetType A,B,C;
     int n;
     printf("创建  种类 1 :\n");
     printf("\t 数量: ");
     scanf("%d", &n);
     creatSet(A,n);
     printf("  种类 1=");
     outputSet(A);
     printf("创建  种类 2 :\n");
     printf("\t 数量: ");
     scanf("%d", &n);
     creatSet(B,n);
     printf("  种类 2=");
     outputSet(B);
     InitSqList(C,listLength(A)+listLength(B)+10);
     setUnion(A,B,C);
     printf("\n 全部种类  =  种类 1  ∪种类 2 =");
     outputSet(C);
     setIntersection(A,B);
     printf("共有种类  =  种类 1  ∩种类 2=");
     outputSet(A);
     return 0;
```

}

3．程序运行结果

```
创建  种类1 :
        数量: 3
        输入3个整数：3 2 1
 种类1={ 3 2 1 }
创建  种类2 :
        数量: 4
        输入4个整数：5 4 3 2
 种类2={ 5 4 3 2 }

全部种类 = 种类1 ∪种类2 ={ 3 2 1 5 4 }
共有种类 = 种类1 ∩种类2={ 3 2 }

--------------------------------
Process exited after 16.08 seconds with return value 0
请按任意键继续. . .
```

成绩：_____