数据结构实验二 顺序表的应用

[ 发布人： [易法令](https://www.scholat.com/flyi)   截止时间：2021-09-23 23:59:59  ]

**柴祥 2030502202 计科202**

一、实验目的

1、掌握建立顺序表的基本方法。

2、掌握顺序表的插入、删除算法的思想和实现，并能灵活运用

二、实验内容

    用顺序表实现病历信息的管理与查询功能。具体要求如下:

1. 利用教材中定义顺序表类型存储病人病历信息(病历号,姓名，症状)；要求使用头文件。

2.设计顺序表定位查找算法，写成一个函数，完成的功能为:在线性表L中查找数据元素x，如果存在则返回线性表中和x值相等的第1个数据元素的序号；如果不存在，则返回-1。

    函数定义为 int ListFind(SequenceList L,char \*x)

请在主函数中测试查找是否存在姓名为x的病人，并根据返回的序号打印出病人信息。

三、实验源代码#include <stdio.h>

#include <string.h>

#define MaxSize 100                                     //宏

#define N 3

typedef struct{

    char num[5];

    char name[10];

    char age[3];

    char sex[5];

    char sym[100];

}patient;

typedef patient ElemType;

#include "SequenceList.h"                                //自定义头文件

void main(){                                            //变量

patient s;

SequenceList myl;

char find[10];

int i,k;

ListInit(&myl);                                          //实现过程

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

    printf("please scanf %d patient's information!---------------\n",i+1);

    printf("please scanf %d patient's num!\n",i+1);scanf("%s",s.num);

    printf("please scanf %d patient's name!\n",i+1);scanf("%s",s.name);

    printf("please scanf %d patient's age!\n",i+1);scanf("%s",s.age);

    printf("please scanf %d patient's sex!\n",i+1);scanf("%s",s.sex);

    printf("please scanf %d patient's symptom!\n",i+1);scanf("%s",s.sym);

    ListInsert(&myl,i,s);

}

printf("\n--------------------datebase----------------\n");

    printf("please scanf what you want to find!\n");

    scanf("%s",find);

    //printf("%s",find);

    for(k=0;k<i;k++){

        ListGet(myl,k,&s); //if(s.num==find   ||  s.name==find   ||  s.age==find   ||  s.sex==find   ||  s.sym==find  )

       if( strcmp(s.num,find)==0  ||  strcmp(s.name,find)==0  || strcmp(s.age,find)==0 ||  strcmp(s.sex,find)==0  || strcmp(s.sym,find)==0)

      {

    printf("%s \n%s \n%s \n%s \n%s \n",s.num,s.name,s.age,s.sex,s.sym);

       }

    }

}

#ifndef SequenceList\_H

#define SequenceList\_H

#include <stdio.h>

#include <stdlib.h>

#include<string.h>

typedef struct{

    ElemType list[MaxSize];

    int size;

}SequenceList;

void ListInit(SequenceList \*L){

    L->size=0;

}

int ListInsert(SequenceList\*L,int i,ElemType x){

    int j;

    if(L->size >=MaxSize){

        printf("Sequencelist  overflow!!\n");

        return 0;

    }

    else if(i<0 ||  i>L->size){

        printf("parametee 'i' is invalid when inserted!\n");

        return 0;

    }

    else{

        for(j=L->size;j>=i;j--){

            L->list[j]=L->list[j-1];

            L->list[i]=x;

            L->size++;

            return 1;

        }

    }

}

int ListGet(SequenceList L ,int i,ElemType \*x){

    if(i<0 || i>L.size){                                                            //?

        printf("parametee 'i' is invalid when getten!");

        return 0;

    }

    else{

        \*x =L.list[i];

        return 1;

    }

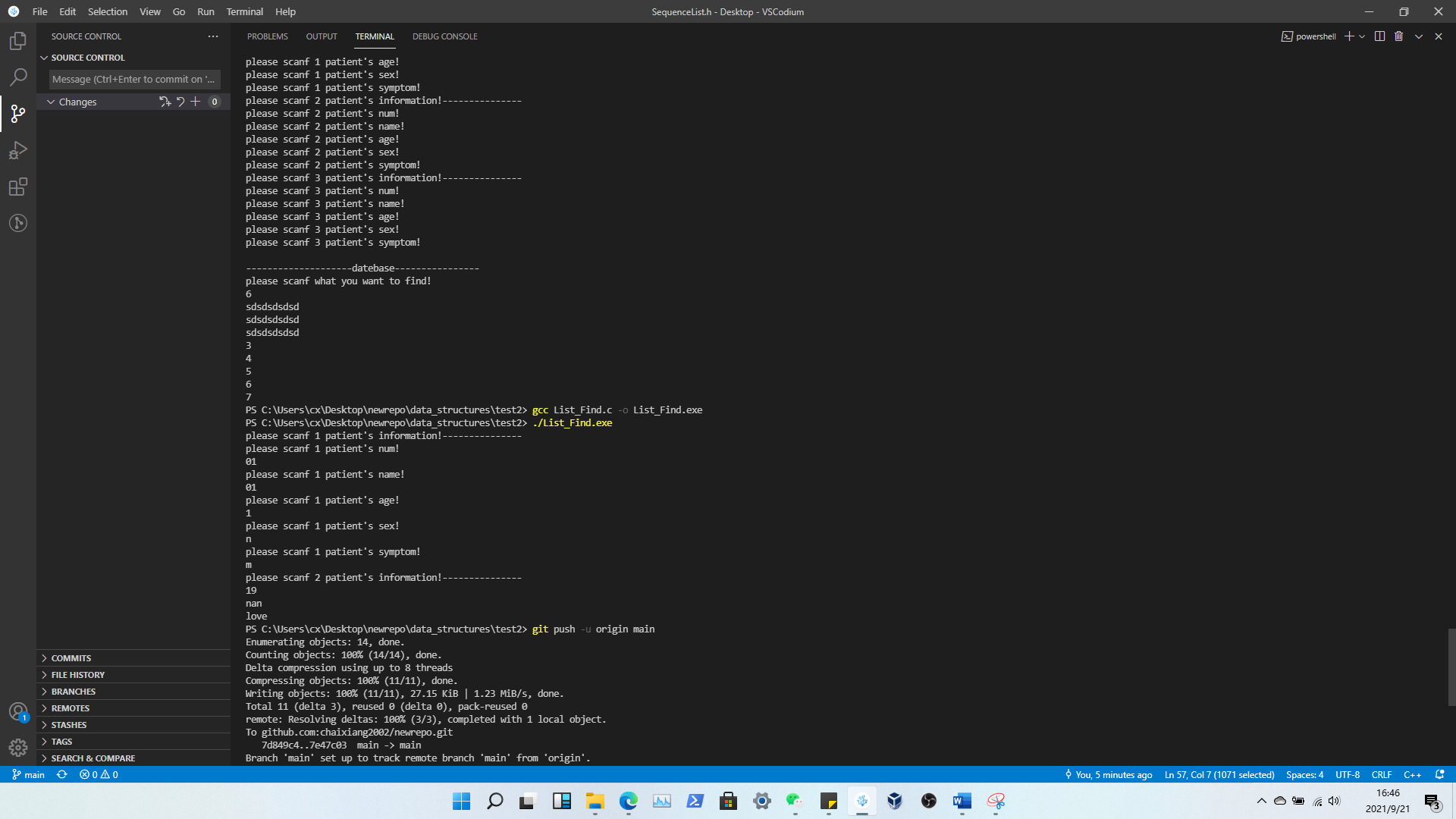
}

int Length(SequenceList L){

    return L.size;

}

#endif

四、实验结果

五、实验心得

1.做的时候有太多问题了

2.就做了一个错误点.md文件在[newrepo/eorror.md at main · chaixiang2002/newrepo (github.com)](https://github.com/chaixiang2002/newrepo/blob/main/data_structures/test2/eorror.md)

个人仓库：git@github.com:chaixiang2002/newrepo.git

3. 