

一、【实验目的】

- 1、掌握建立单链表的基本方法。
- 2、掌握单链表的插入、删除算法的思想和实现

二、【实验内容】

仿照教材中的单链表实现例子，自己设计一个有序单链表，单链表中的数据元素为整型并递增有序。有序单链表的定义：

逻辑结构:有序线性表，数据元素递增有序

存储结构:链式

操作集合:初始化、插入、删除、撤销

(1)ListInitiate(L) 初始化线性表，生成一个空表 L。

(2)ListInsert(L,x) 在有序表 L 中插入数据元素 x,使得新表仍然有序。

(3)ListDelete(L,x) 删除有序表 L 中的数据元素 x，若删除成功则返回 1，不成功则返回 0。

(4)Destroy(L) 撤销单链表

要求:

- 1.有序单链表的操作集合有如下操作:初始化、插入、删除、撤销，使用头文件单链表的代码。
- 2.编写主函数 main()验证所设计的有序单链表是否能正确插入、删除。

提示:

- 1.插入操作时，从链表的第一个数据元素结点开始，逐个比较每个结点的 data 域值和 x 的值，当 data 小于等于 x 时，进行下一个结点的比较；否则就找到了插入结点的合适位置，此时申请新结点把 x 存入，然后把新结点插入；当比较到最后一个结点仍有 data 小于等于 x 时，则把新结点插入单链表尾。
- 2.删除操作时，从链表的第一个数据元素结点开始，逐个比较每个结点的 data 域值和 x 的值，当 data 不等于 x 时，进行下一个结点的比较；否则就找到了要删除的结点，删除结点后释放结点。如果到了表尾还没有找到值为 x 的结点，则链表中没有要删除的元素。

三、实验源代码

```
#ifndef LinkedList_H
#define LinkedList_H
```

```

#include <stdio.h>
#include <stdlib.h>
#include<string.h>
typedef int ElemType;
typedef struct SingleNode{ //????
ElemType date;
struct SingleNode *next;
}SingleLinkedList,*LinkedList;

void ListInit(SingleLinkedList **head){
if((*head=(SingleLinkedList*)malloc(sizeof(SingleLinkedList)))==NULL){exit(1);}
(*head)->next=NULL;
}

int Lenth(SingleLinkedList *head){
SingleLinkedList *p=head;
int size=0;
while (p->next!=NULL) {
p=p->next;
size++;
}
// printf("size=%d\n",size);
return size;
}

int ListGet(SingleLinkedList *head,int i,ElemType *x){
// printf("sdsada\n");
SingleLinkedList *p;
int j;
p=head;
j=-1;
while (p->next!=NULL&& j<i) {
p=p->next;j++;
}
if(j!=i){
printf("Error taking element positon argument");
return 0;
}
*x=p->date;
return 1;
}

int ListInsert(SingleLinkedList *head,int i,ElemType x){ //小-->>大
SingleLinkedList*p,*q;
int j;
p=head;
j=-1;

```

```

while (p->next!=NULL && x>p->date && j<i-1) {
p=p->next;
j++;
}
if (j!=i-1) {
printf("insert position parameter error!");
return 0;
}
if((q=(SingleLinkedList *)malloc(sizeof(SingleLinkedList)))==NULL)
{exit(1);}
q->date=x;
q->next=p->next;
p->next=q;
return 1;
}

```

```

int ListDelete(SingleLinkedList *head,int i,ElemType *x){
SingleLinkedList *p,*s;
int j;
p=head;
j=-1;
while (p->next!=NULL&&p->next->next!=NULL&&j<i-1) {
p=p->next;
j++;
}
if(j!=i-1){
printf("insert position parameter 12error!");
}
s=p->next;
*x=s->date;
p->next=s->next;
free(s);
return 1;
}

```

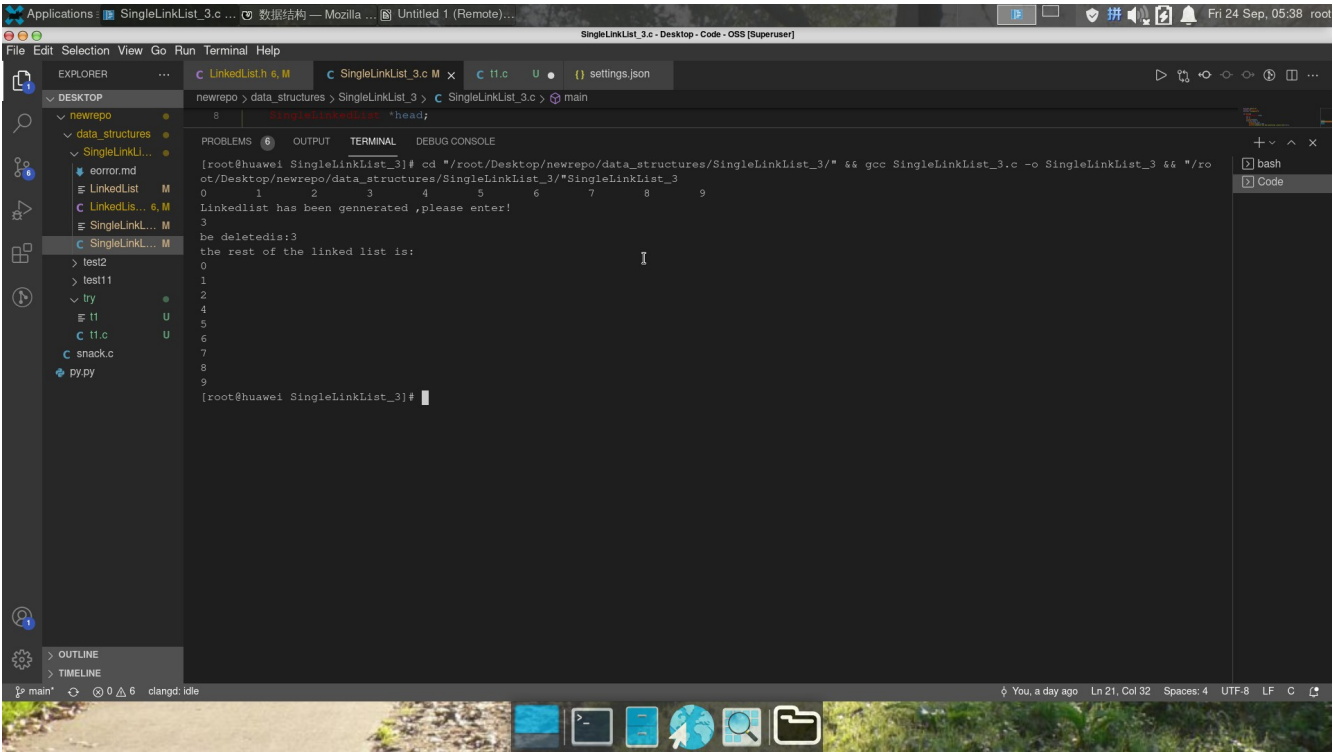
```

void ListDestroy(SingleLinkedList **head){
SingleLinkedList *p,*p1;
p=*head;
while (p!=NULL) {
p1=p;
p=p->next;
free(p1);
}
*head=NULL;
}

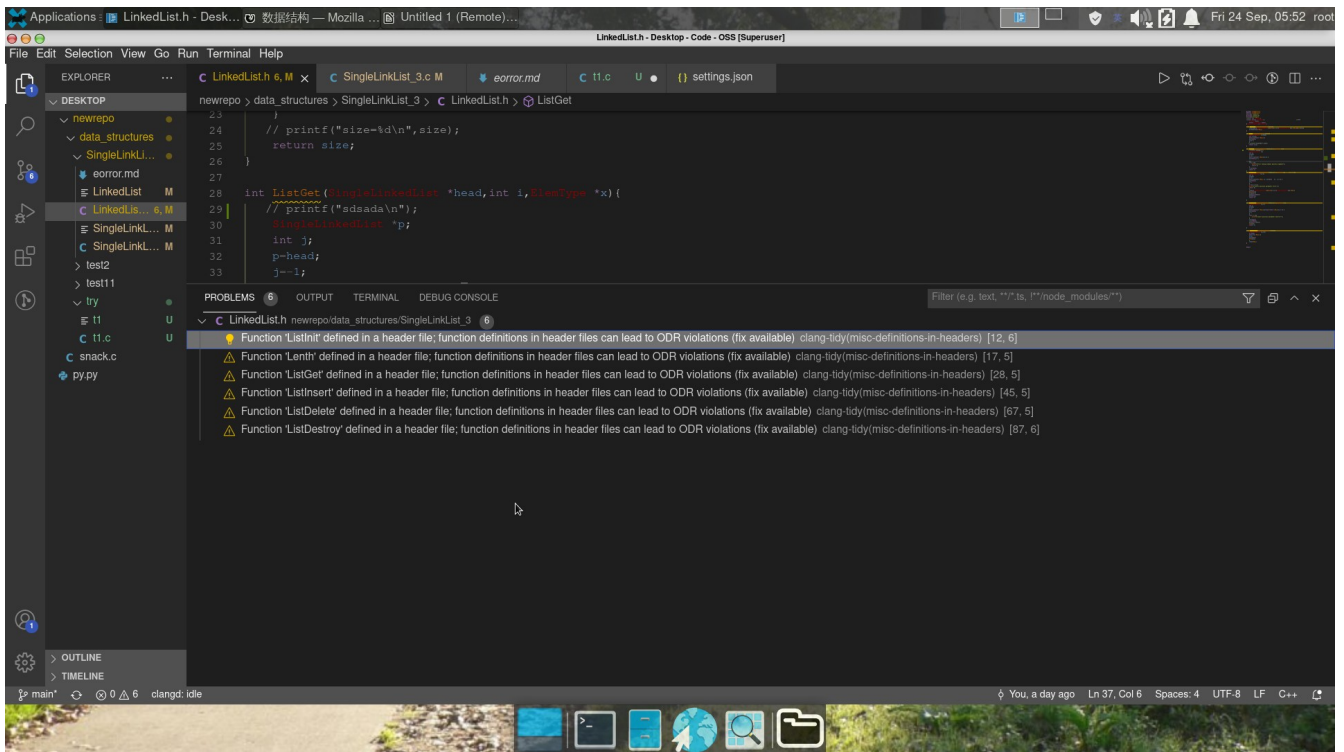
```

#endif

四、实验结果



五、实验心得



为什么他要让我用内联函数？？？？？
累死了