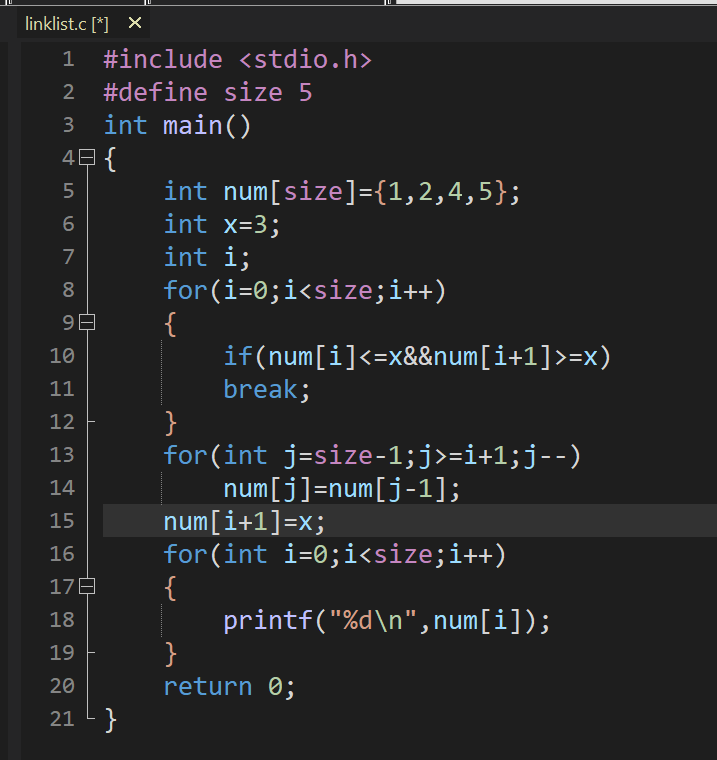
作业二

1. 算法一：逐个比较

for(i=0;i<L.length;i++)

if(L.elem[i]<=x&&L.elem[i+1]>=x) ListInsert(&L,i+1,x);

时间复杂度为O（n^2）





算法二：简要思路为近似二分查找

int left=0;

int right=elenum;

while(

if(L.elem[(left+right)/2]>x) right=(left+right)/2;

else if(L.elem[left+right)/2<x) left=(left+right)/2;

else ListInsert(&L,(left+right)/2,x);

1. 算法一:逐个比较

for(i=0;i<L.length;)

if(LocateElem(L,L.elem[i])==0) i++ ;

else ListDelete(&L,LocateElem(L,L.elem[i]));

1. 算法一：逐个比较

for(i=0;i<L.length;i++)

if(L.elem[i]>=x&&L.elem[i]<=y) ListDelete(&L,i);

4.算法一：使用两个指针，左边指针往后走，右边指针往前走，如果当前位置是字母，那就和左边位置换，并将左边指针+1，如果是数字，和右边指针换，右边+1

left = 0

right = L.length- 1

current = 0

while current <= right

if isAlphabetic(L[current])

swap(L[current], L[left])

left = left + 1

current = current + 1

else if isNumeric(L[current])

swap(L[current], L[right])

right = right - 1

else

current = current + 1

5.第一种算法：使用min{m,n}的辅助空间

InitList(&L1);

If(m>n)

For(i=0;i<n;i++)

L1.elem[i]=bi;

For(i=m-1;i>=0;i--)

L.elem[i+n]=L.elem[i];

For(i=0;i<n;i++)

L,elem[i]=L1.elem[i]

M<n时同理

第二种算法：使用1的空间

假如n小于m，那么则将b1移到辅助空间，将a1放到b1原来的位置，把b1放到a1的位置，如此交换。可以将全部的b换好，然后如法炮制把a全部换好。