一、实验目的  
1、掌握递归原理

2、掌握一些常用问题的递归算法设计

二、实验内容  
1.编写一个连加运算递归函数 ADD(n)，实现从1+2+……+n。

2. 编写折半查找算法的递归实现和非递归实现。

 提示:将要查找的元素key与查找区间正中元素相比，若key小，则查找区间缩小至前半部份查找，若key大，则查找区间缩小至后半部份查找；再取其中值比较，每次缩小1/2的范围，直到查找成功或失败为止。如递归实现，考虑函数的参数应有哪些。在用循环结构实现时，函数的参数有什么变化?

3.因子分解问题。大于1的正整数n可以分解为:n=x1\*x2\*…\*xm。 例如，当n=12 时，共有8 种不同的分解式: 12=12； 12=6\*2； 12=4\*3； 12=3\*4； 12=3\*2\*2； 12=2\*6； 12=2\*3\*2； 12=2\*2\*3。 对于给定的正整数n，计算n共有多少种不同的分解式。

**三、【**实验源代码**】**

Main.c

#include<stdio.h>

#define max 10

#include"Recursion.h"

int main(){

    int n,m,a[max],b[max],i,j,tmp ;

    printf("问题1：输入ADD算法的n值\n");

    scanf("%d",&n);

    printf("答案是%d",ADD(n));

    int num = printf("输入你在查询算法中查的数") ;

    printf("二分法的递归实现,请输入你数组的值,10个, ,必须是递增的！！\n");

    for(i = 0;i<max;i++){

        scanf("%d",&a[i]);

    }

    tmp = Trecursion(a,0,max,num);

    if(tmp != -1)

    printf("查询成功\n");

    else

    printf("查询失败\n");

    printf("二分法的非递归实现，请输入你数组的值,10个\n");

    for(j=0;j<max;j++){

        scanf("%d",&b[j]);

    }

    tmp = Frecursion(b,num);

    if(tmp == 1)

    printf("查询成功\n");

    else

    printf("查询失败\n");

    printf("问题3：输入Break算法的n值\n");

    scanf("%d",&m);

    printf("答案是%d\n",Break(m));

    return 0;

}

**Recursion.h**

int ADD(int n){

    if(n==1){

        return 1;

    }

    else{

        return n+ADD(n-1);

    }

}

//二分非递归

int Frecursion(int \*a,int key){

    int i = 0,j = max-1;

    //两个头尾指针

    int middle = (i+j)/2;

    while(a[middle] != key){

        if(a[middle] > key){

            j = middle;

            middle = (i+j)/2;

        }

        else if(a[middle] < key)

        {

            i = middle;

            middle = (i+j)/2;

        }

        else

            return 1;

    }

if(a[(i+j)/2+1]  == key )

return 1;

else

return 0;

}

//二分递归

int Trecursion(int \*arr, int low, int high, int target){

    if (low <= high) {

        int mid = low + (high - low) / 2;

        if (arr[mid] == target) {

            return mid;

        } else if (arr[mid] > target) {

            return Trecursion(arr, low, mid - 1, target);

        } else {

            return Trecursion(arr, mid + 1, high, target);

        }

    return -1;

}

}

// 因子分解问题,递归解决,!!!!!!!!

int Break(int n)

{

    int  ans = 1,i;//ans==1是直接让他本身\*1做一种情况

    for (i = 2;i\*i<n;i++)

    {

        if(n%i == 0)

            ans += Break(i) + Break(n/i);

        if(i+i == n)

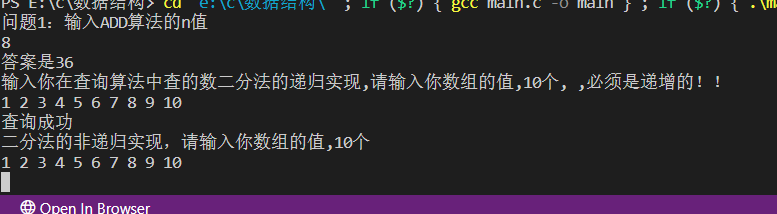
            ans += Break(i);

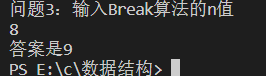
        return ans;

    }

 }

**四、【**实验结果**】**





**五、【**实验心得**】**

 1感觉非递归比递归难写

https://www.scholat.com/images/backTopBottom.png