第三次作业

A卷：

第1题：

#include<stdio.h>

void change(int\* arr, int size);

int main()

{

    int number;

    scanf("%d",&number);

    int arr[number];

    //遍历输入

    for (int i = 0; i < number; i++)

    {

        scanf("%d",&arr[i]);

    }

    change(arr,number);

    //打印出来

    for (int i = 0; i < number; i++)

    {

        printf("%d ",arr[i]);

    }

    return 0;

}

void change(int\* arr, int size)

{

    for (int i = 0; i < size; i++)

    {

        //如果是负数，则将负数转化为正数

        if (arr[i] < 0)

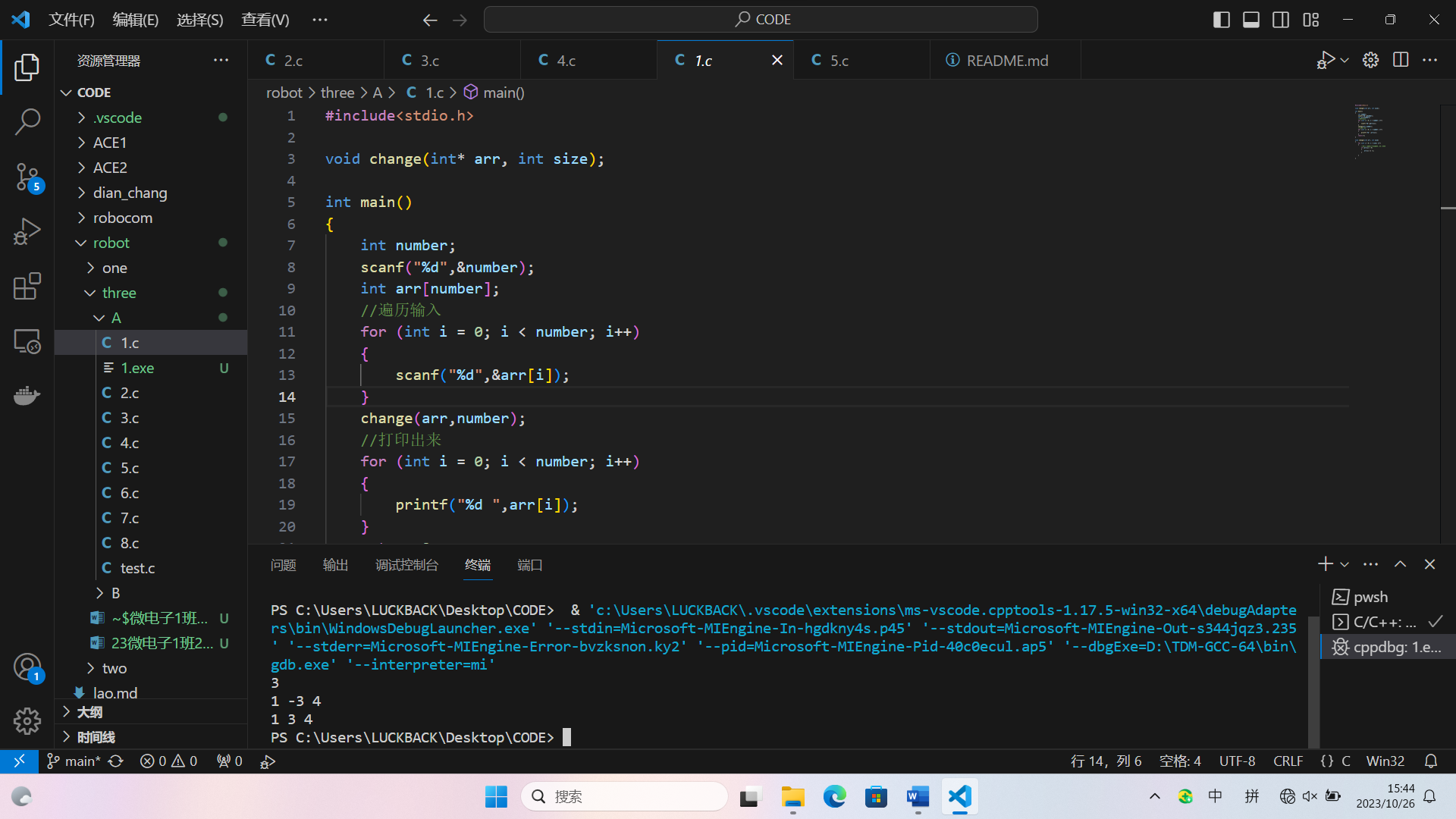
        {

            arr[i] \*= -1;

        }

    }

}



第2题：

#include<stdio.h>

//建立结构模板

struct Triangle

{

    int a;

    int b;

    int c;

};

int main()

{

    //创建结构类型

    struct Triangle apple = {

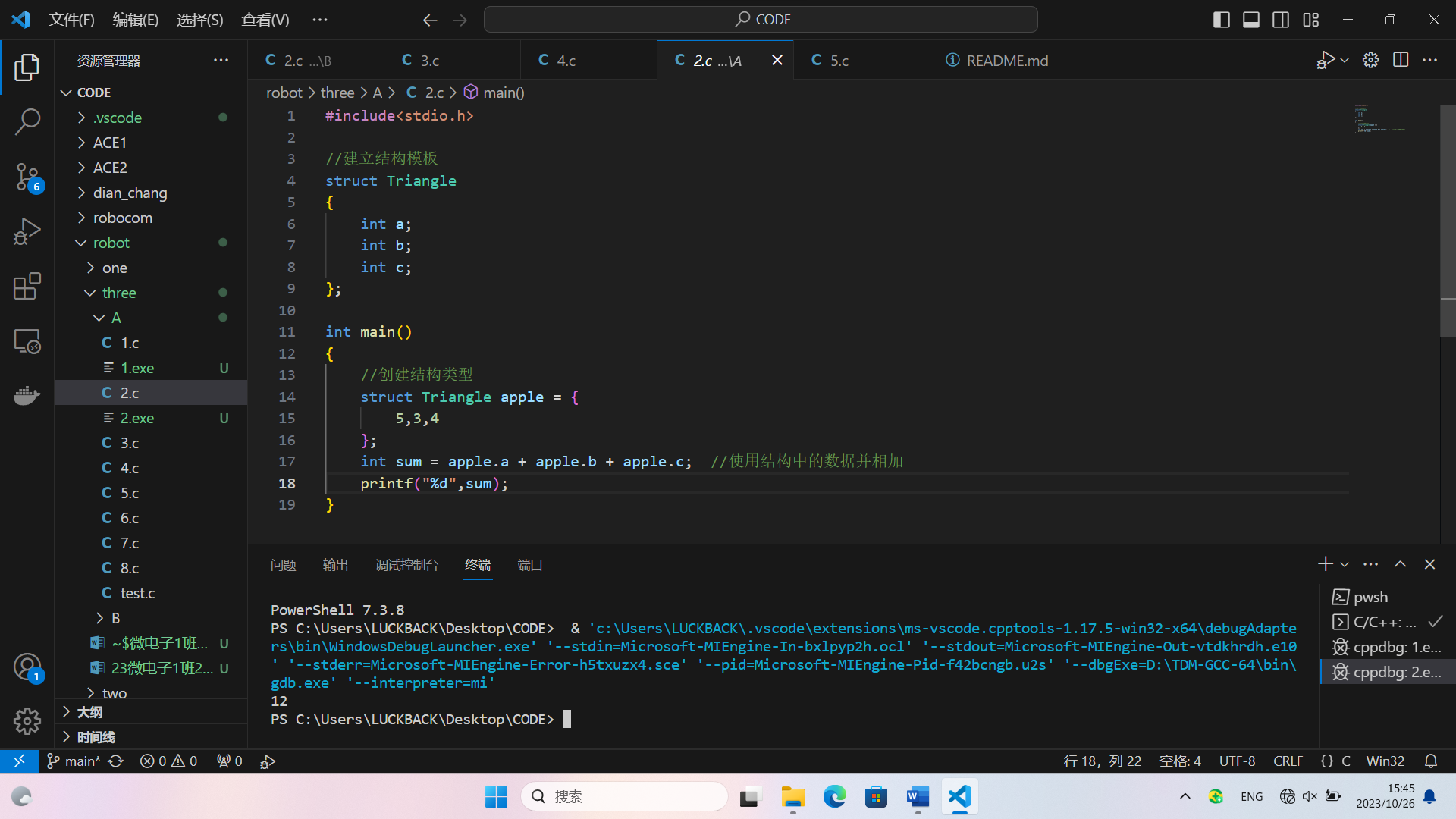
        5,3,4

    };

    int sum = apple.a + apple.b + apple.c;  //使用结构中的数据并相加

    printf("%d",sum);

}



第3题：

#include<stdio.h>

int main()

{

    int arr[20] = {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20};

    int \*p; //建立指向数组元素的指针变量

    for (int i = 19; i >= 0; i--)

    {

        p = &arr[i];    //将数组元素的地址给指针

        printf("%d ",\*p);   //打印指针指向地址所表示的值

        //每五个换行

        if (i % 5 == 0)

        {

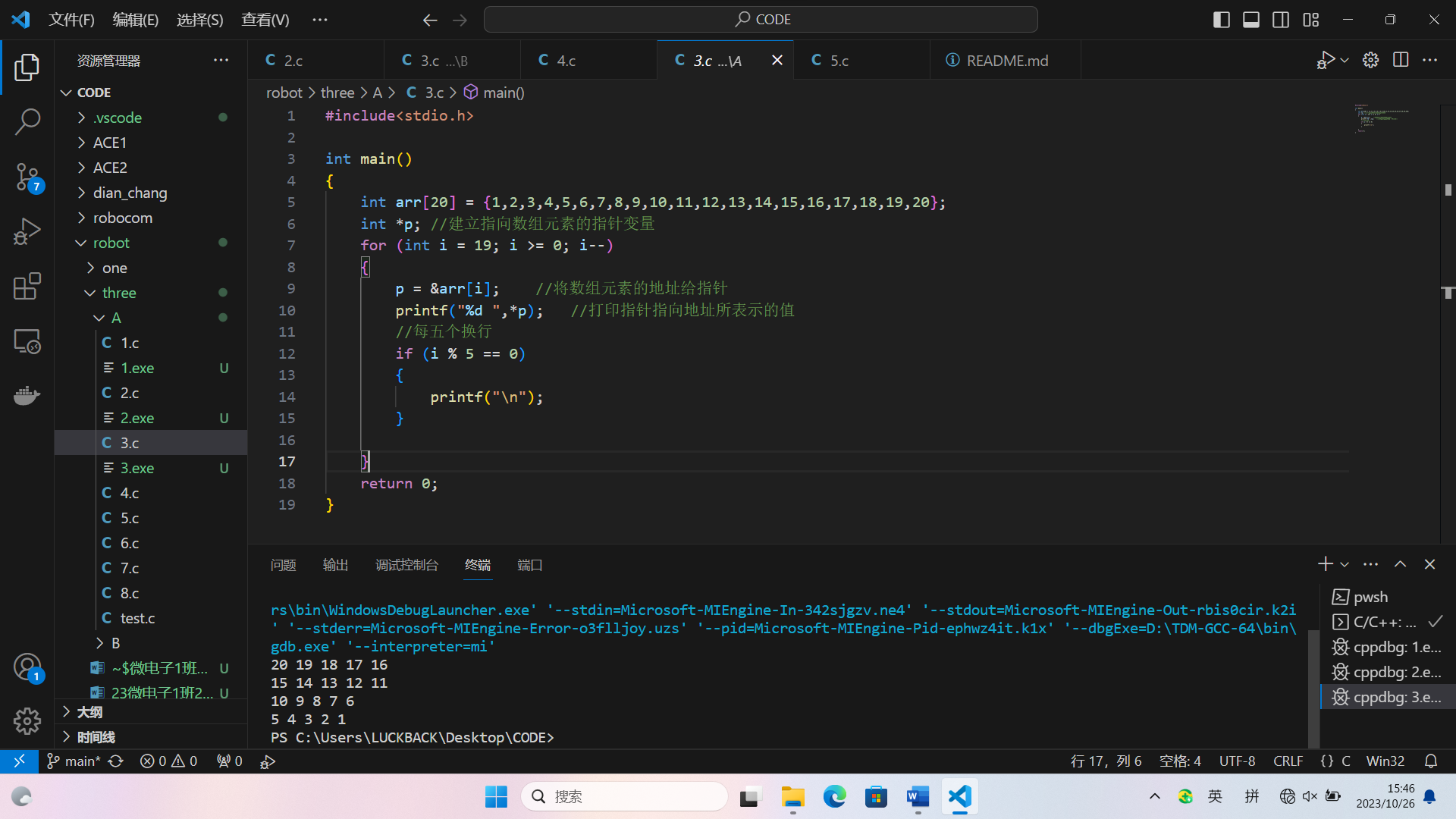
            printf("\n");

        }

    }

    return 0;

}



第4题：

#include<stdio.h>

int exchange(int number);

int main()

{

    int number;

    printf("输入一个正整数:\n");

    scanf("%d",&number);

    int ex = exchange(number);

    printf("十进制数%d转化为二进制为:\n%d",number,ex);

}

//十进制转换

int exchange(int number)

{

    int two = 0;

    //根据十进制转换为二进制

    /\*    5/2=2...1

          2/2=1...0

          1/2=0...1

          5的二进制为101\*/

    for(int i = 1; i < 10000000; i \*= 10)

    {

        int unit = number%2;

        number /= 2;

        if(unit == 1)

        {

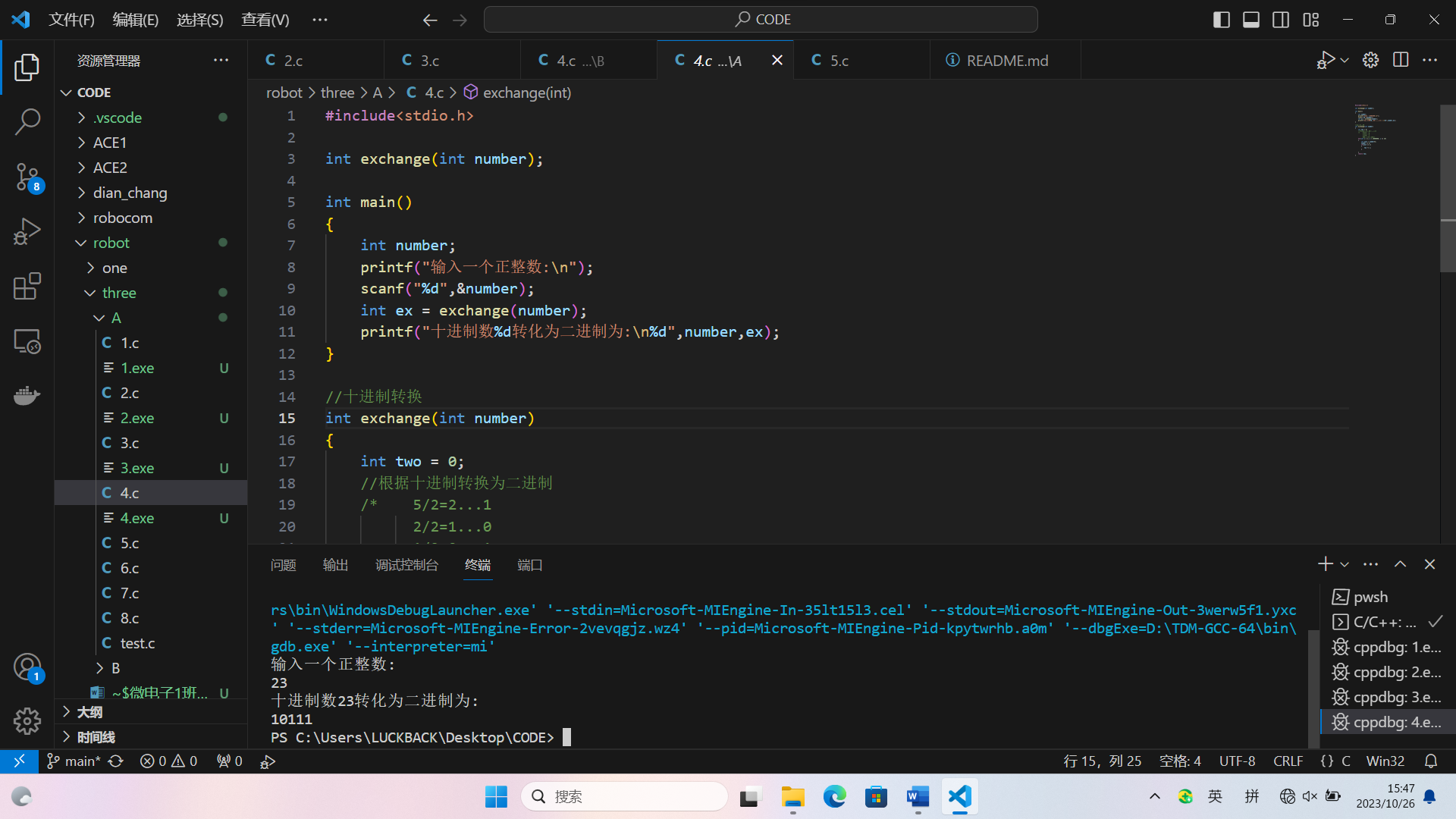
            two += i;

        }

    }

    return two;

}



第5题：

#include<stdio.h>

#include<stdbool.h>

//创建结构模板

struct Triangle

{

    int a;

    int b;

    int c;

};

bool is\_right(struct Triangle\* tr);

int main()

{

    struct Triangle tr;

    scanf("%d %d %d",&tr.a,&tr.b,&tr.c);

    bool isRight = is\_right(&tr);

    //判断

    if (isRight)

    {

        printf("是直角三角形。");

    }

    else

    {

        printf("不是直角三角形。");

    }

    return 0;

}

bool is\_right(struct Triangle\* tr)

{

    bool is\_right\_triangle = false;

    //判断是否为直角三角形

    if (tr->a \* tr->a + tr->b \* tr->b == tr->c \* tr->c ||

        tr->a \* tr->a + tr->c \* tr->c == tr->b \* tr->b ||

        tr->c \* tr->c + tr->b \* tr->b == tr->a \* tr->a)

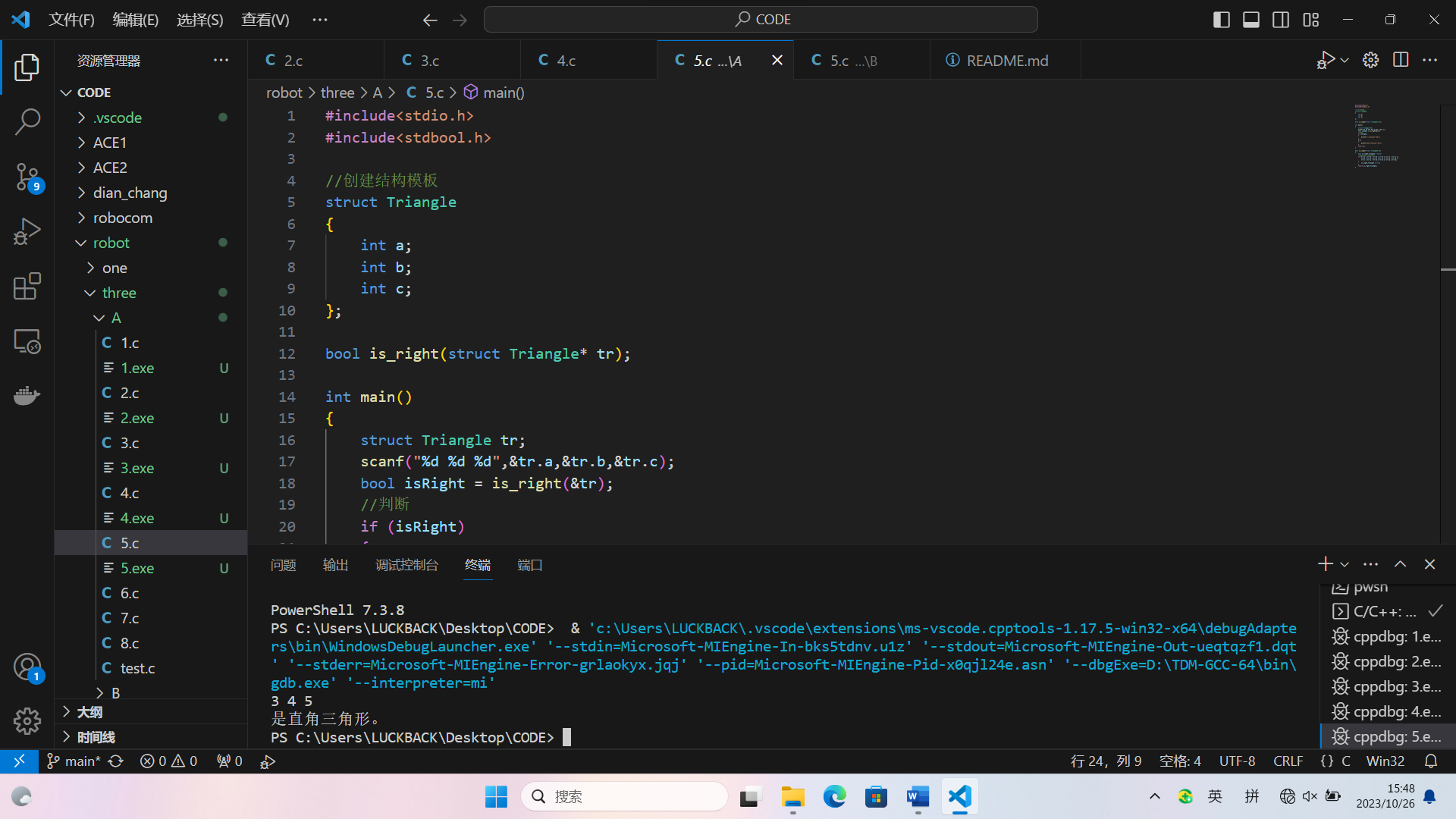
    {

        is\_right\_triangle = true;

    }

    return is\_right\_triangle;

}



第6题：

#include<stdio.h>

#include<stdbool.h>

#include<string.h>

struct Student

{

    int id;  // 学号

    char name[32]; // 姓名

    bool sex; // 性别

};

void st\_set(struct Student\* st, int id, char\* name, bool sex);

int main()

{

    struct Student stu;

    int id = 2023;

    char name[20] = "马一鸣";

    bool sex = true;

    st\_set(&stu,id,name,sex);

    printf("%d %s %d",id,name,sex);

}

void st\_set(struct Student\* st, int id, char\* name, bool sex)

{

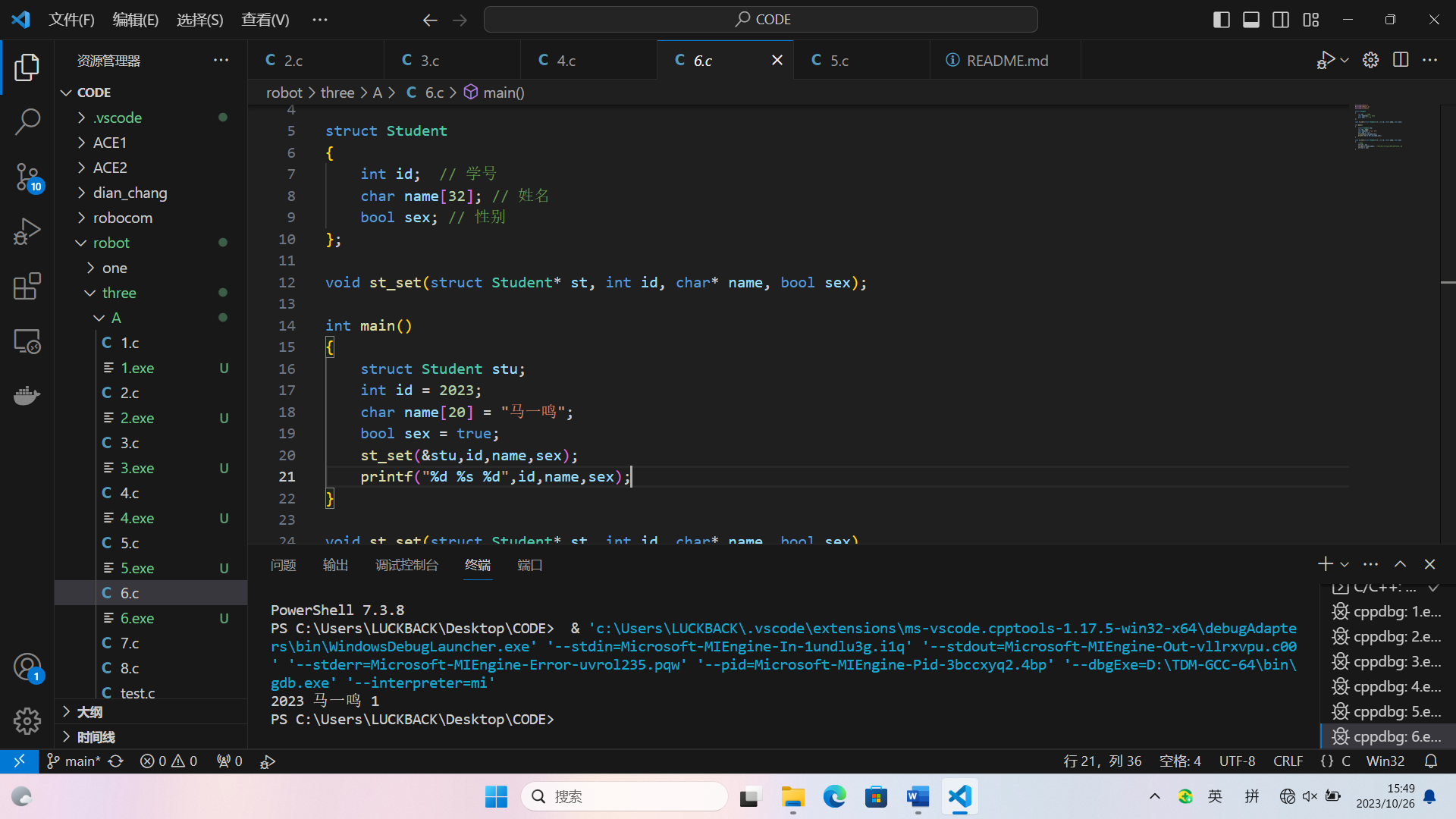
    //这样

    st->id = id;

    strcpy(st->name,name);  //这里要用strcpy函数，否则我不行。

    st->sex = sex;

}



第7题：

#include<stdio.h>

struct TrafficLight

{

    int color; // 0, 表示红色， 1 表示黄 ， 2表示绿

};

void change(struct TrafficLight\* light);

void show(struct TrafficLight\* light);

int main()

{

    struct TrafficLight traff;

    traff.color = 0;    //初始化为红灯

    change(&traff);

    show(&traff);

}

void change(struct TrafficLight\* light)

{

    //红灯变绿灯

    if (light->color == 0)

    {

        light->color = 2;

    }

    //绿灯变黄灯

    else if (light->color == 2)

    {

        light->color = 1;

    }

    //黄灯变红灯

    else if (light->color == 1)

    {

        light->color == 0;

    }

}

void show(struct TrafficLight\* light)

{

    //红灯

    if (light->color == 0)

    {

        printf("红灯");

    }

    //绿灯

    else if (light->color == 2)

    {

        printf("绿灯");

    }

    //黄灯

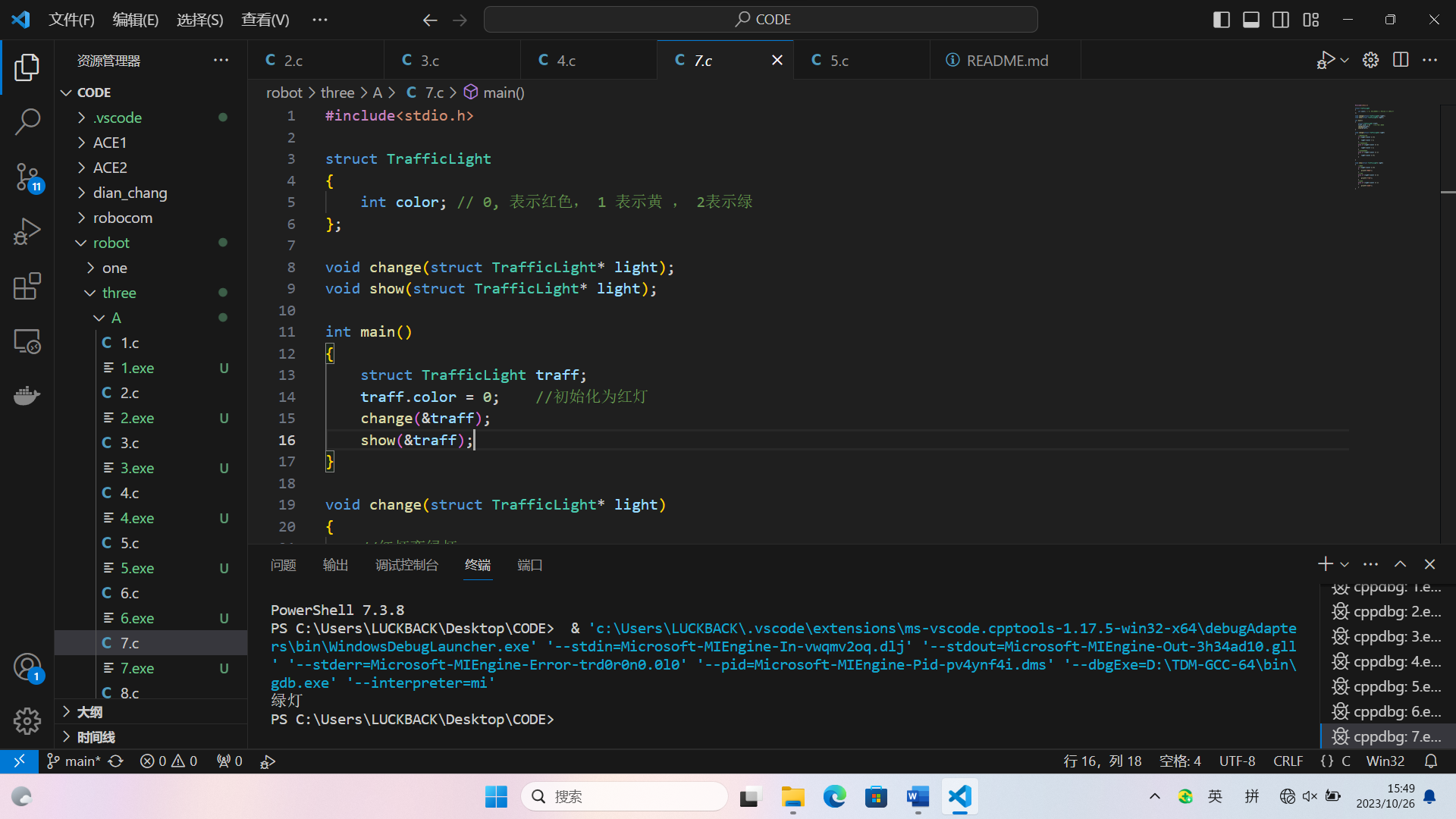
    else if (light->color == 1)

    {

        printf("黄灯");

    }

}



第8题：

#include<stdio.h>

struct Fraction

{

    int num; // 分子

    int den; // 分母

};

struct Fraction add (struct Fraction a,  struct Fraction b);    //加法

struct Fraction mul (struct Fraction a,  struct Fraction b);    //减法

struct Fraction Simplify(struct Fraction a);                    //化简

int Least\_common\_multiple(int a,int b);                         //最小公倍数

int greatest\_common\_divisor(int a,int b);                       //最大公约数

int main(void)

{

    struct Fraction fra1,fra2,addition,mulit;

    //输入

    scanf("%d/%d",&fra1.num,&fra1.den);

    scanf("%d/%d",&fra2.num,&fra2.den);

    //调用函数

    addition = add(fra1,fra2);

    mulit = mul(fra1,fra2);

    printf("%d/%d\n",addition.num,addition.den);

    printf("%d/%d",mulit.num,mulit.den);

    return 0;

}

struct Fraction add (struct Fraction a,  struct Fraction b)

{

    struct Fraction addition;

    //如果分母相等，直接让分子相加就行

    if (a.den == b.den)

    {

        addition.num = a.num + b.num;

        addition.den = a.den;

    }

    //如果不相等就先同分再相加

    else

    {

        int multiple = Least\_common\_multiple(a.den,b.den);

        addition.num = a.num \* (multiple / a.den) + b.num \* (multiple / b.den);

        addition.den = multiple;

    }

    addition = Simplify(addition);

    return addition;

}

struct Fraction mul (struct Fraction a,  struct Fraction b)

{

    struct Fraction mulit;

    //如果分母相等，直接让分子相减就行

    if (a.den == b.den)

    {

        mulit.num = a.num - b.num;

        mulit.den = a.den;

    }

    //如果不相等就先同分再相减

    else

    {

        int multiple = Least\_common\_multiple(a.den,b.den);

        mulit.num = a.num \* (multiple / a.den) - b.num \* (multiple / b.den);

        mulit.den = multiple;

    }

    mulit = Simplify(mulit);

    return mulit;

}

//还要求出最小公倍数

int Least\_common\_multiple(int a,int b)

{

    //将min等于最大的那个数

    int min = a;

    if (b > min)

    {

        min = b;

    }

    //将每一个都试一试

    while(1)

    {

        if (min % a == 0 && min % b ==0)

        {

            break;  //找出来才能退出循环

        }

        min++;

    }

    return min;

}

//我觉得还是要化简一下

struct Fraction Simplify(struct Fraction a)

{

    int min = greatest\_common\_divisor(a.den,a.num);

    a.den /= min;

    a.num /= min;

    return a;

}

//求出最大公约数

int greatest\_common\_divisor(int a,int b)

{

    //让min为最小的那个数

    int min = a;

    if (a < min)

    {

        min = b;

    }

    //一直遍历到一，直到找到为止

    while (min > 1)

    {

        if (a % min == 0 && b % min == 0)

        {

            break;

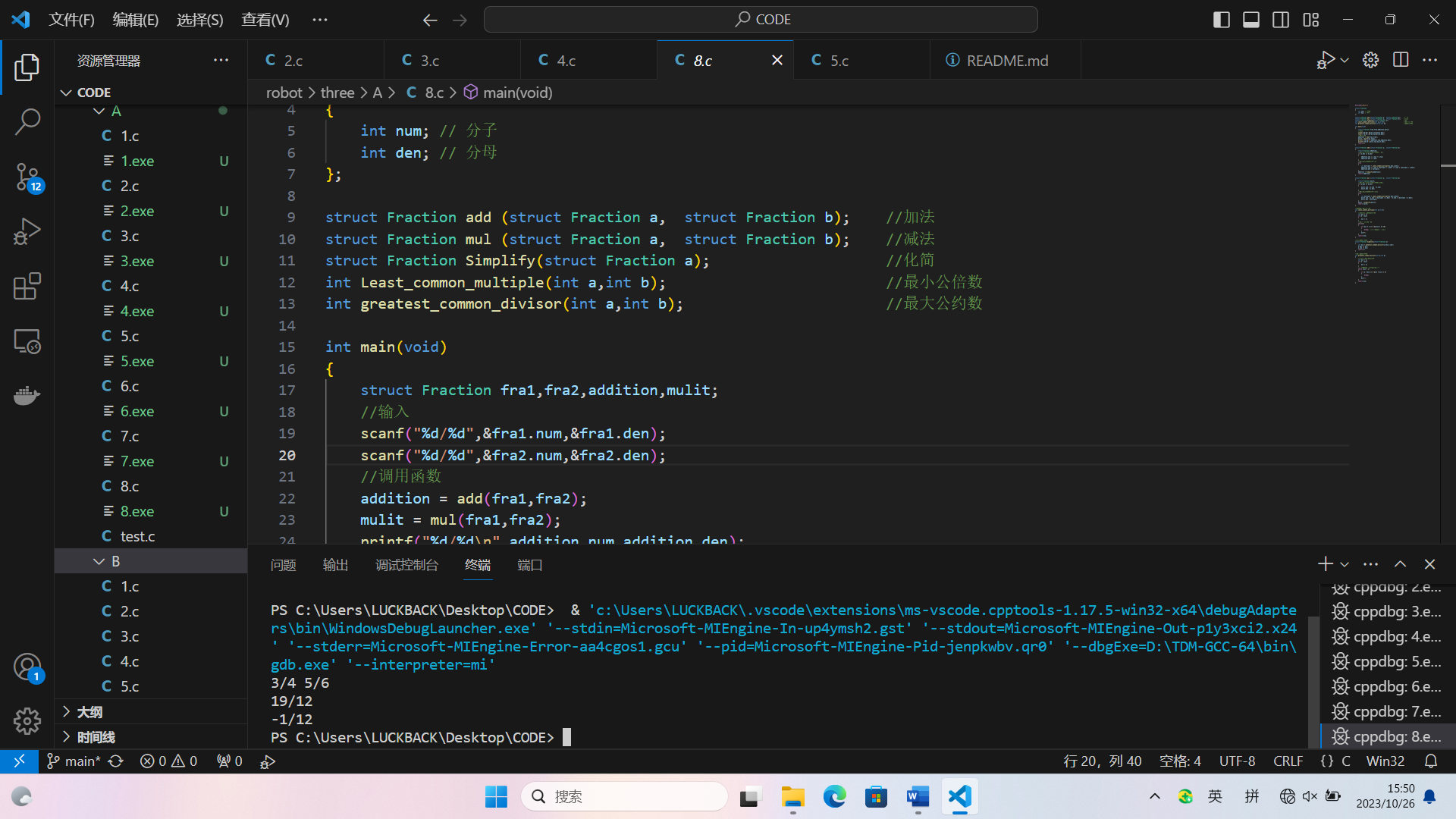
        }

        min--;

    }

    return min;

}



B卷：

第1题：

这道题还没办法转成十六进制

#include<stdio.h>

#include<math.h>

void Hexadecimal\_conversion(int number,int origin\_hexadecimal,int exchange\_hexadecimal);

int exchange(int number,int exchange\_hexadecimal);

int main()

{

    int origin\_Hexadecimal,number,exchange\_hexadecimal;

    scanf("%d %d %d",&number,&origin\_Hexadecimal,&exchange\_hexadecimal);

    Hexadecimal\_conversion(number,origin\_Hexadecimal,exchange\_hexadecimal);

    return 0;

}

//先转化为十进制，再转化为其他进制

void Hexadecimal\_conversion(int number,int origin\_hexadecimal,int exchange\_hexadecimal)

{

    int ten\_number = 0;

    int digit = 0;

    while (number)

    {

        int unit = number % 10;

        number /= 10;

        ten\_number += unit \* pow(origin\_hexadecimal,digit);

        digit++;

    }

    printf("%d\n",ten\_number);

    int exchange\_number = exchange(ten\_number,exchange\_hexadecimal);

    printf("%d",exchange\_number);

}

//十进制转换

int exchange(int number,int exchange\_hexadecimal)

{

    int exchange\_number = 0;

    //根据十进制转换为二进制为例

    /\*    5/2=2...1

          2/2=1...0

          1/2=0...1

          5的二进制为101\*/

    for(int i = 1;number != 0;i \*= 10)

    {

        int unit = number%exchange\_hexadecimal;

        number /= exchange\_hexadecimal;

        if(unit)

        {

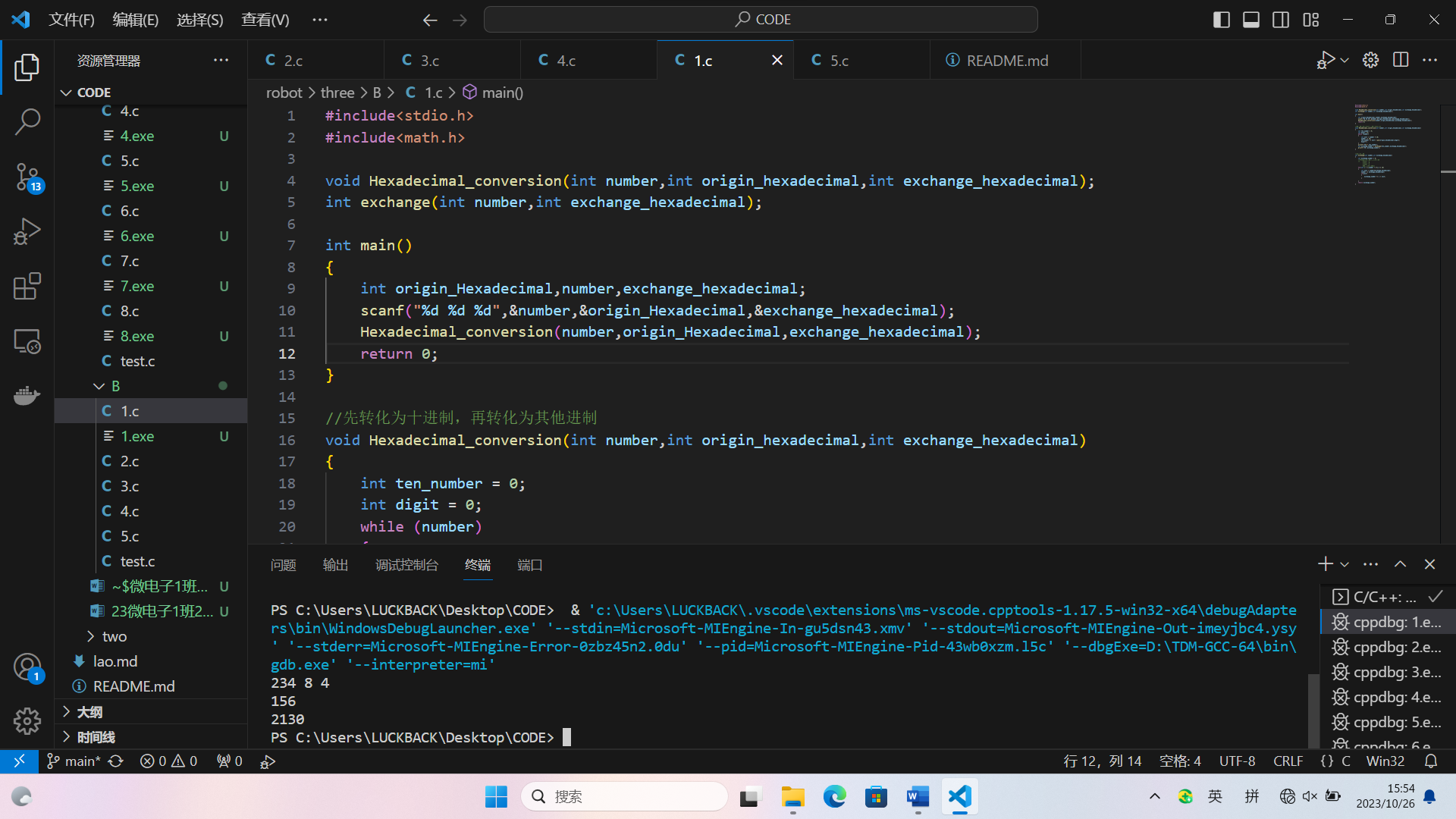
            exchange\_number += i \* unit;

        }

    }

    return exchange\_number;

}



第2题：

#include<stdio.h>

void exchange(int number,char one,char two,char three);

void move(char origin,char change);

//我好像只能这样设立全局变量

int step = 0;

int main()

{

    int n;

    scanf("%d",&n);

    exchange(n,'A','B','C');

    printf("%d",step);  //2的n次方减一

    return 0;

}

//number表示的是要转移几个,one表示从哪里转移,two表示中转站是哪个,three表示要移动到哪里

void exchange(int number,char one,char two,char three)

{

    if (number == 1)

    {

        move(one,three);

    }

    //如同书上的一样，先将n-1个放在中间，再将最后一个放在右边，并以此类推。

    else

    {

        exchange(number-1,one,three,two);

        move(one,three);

        exchange(number-1,two,one,three);

    }

}

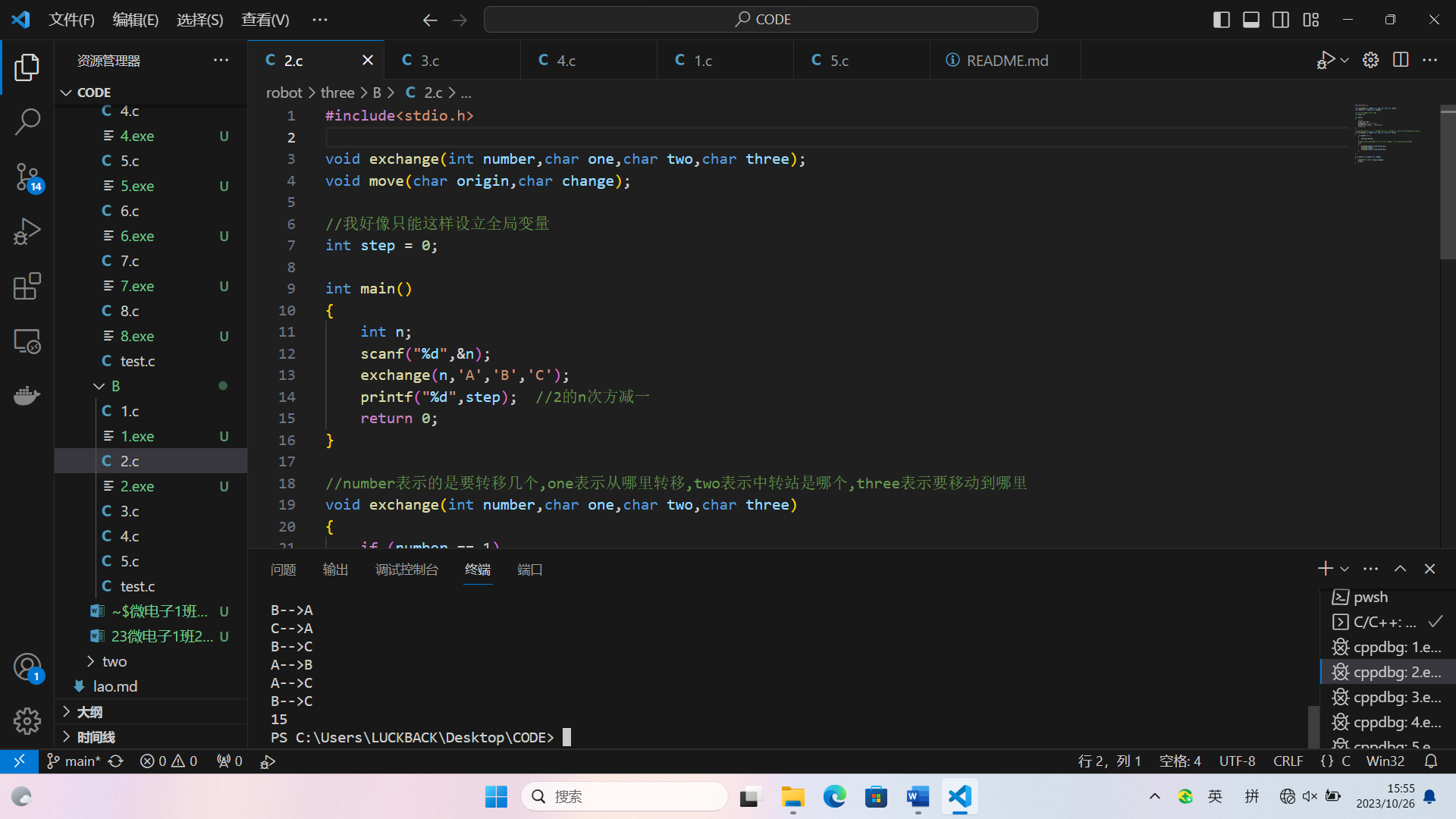
void move(char origin,char change)

{

    printf("%c-->%c\n",origin,change);

    step++;

}



第3题：

#include<stdio.h>

struct Gun

{

    char\* name;

    int damage;

};  //千万要注意这个;号。

struct Person

{

    char \*name;

    int hp;

};

struct Gun CreatGun(char\* name,int damage);

struct Person CreatePerson(char\* name,int hp);

//Person1为攻击的人，Person2为被攻击的人，gun为拿什么枪打的

void shoot(struct Person \*person1 , struct Person \*person2,struct Gun \*gun);

int main(void)

{

    struct Gun AK\_47 = CreatGun("AK-47",10);

    struct Gun AWP = CreatGun("AWP",100);

    struct Person person1 = CreatePerson("马上回来",100);

    struct Person person2 = CreatePerson("苏秋白",100);

    shoot(&person1,&person2,&AK\_47);

    shoot(&person1,&person2,&AK\_47);

    shoot(&person1,&person2,&AK\_47);

    shoot(&person2,&person1,&AWP);

    return 0;

}

struct Gun CreatGun(char\* name,int damage)

{

    struct Gun the\_gun = {

        name,

        damage

    };

    return the\_gun;

}

struct Person CreatePerson(char\* name,int hp)

{

    struct Person the\_person ={

        name,

        hp

    };

    return the\_person;

}

//Person1为攻击的人，Person2为被攻击的人，gun为拿什么枪打的

void shoot(struct Person \*person1 , struct Person \*person2,struct Gun \*gun)

{

    person2->hp -= gun->damage;

    if (person2->hp <= 0)

    {

        printf("%s已死亡,被%s使用%s杀死。\n",person2->name,person1->name,gun->name);

    }

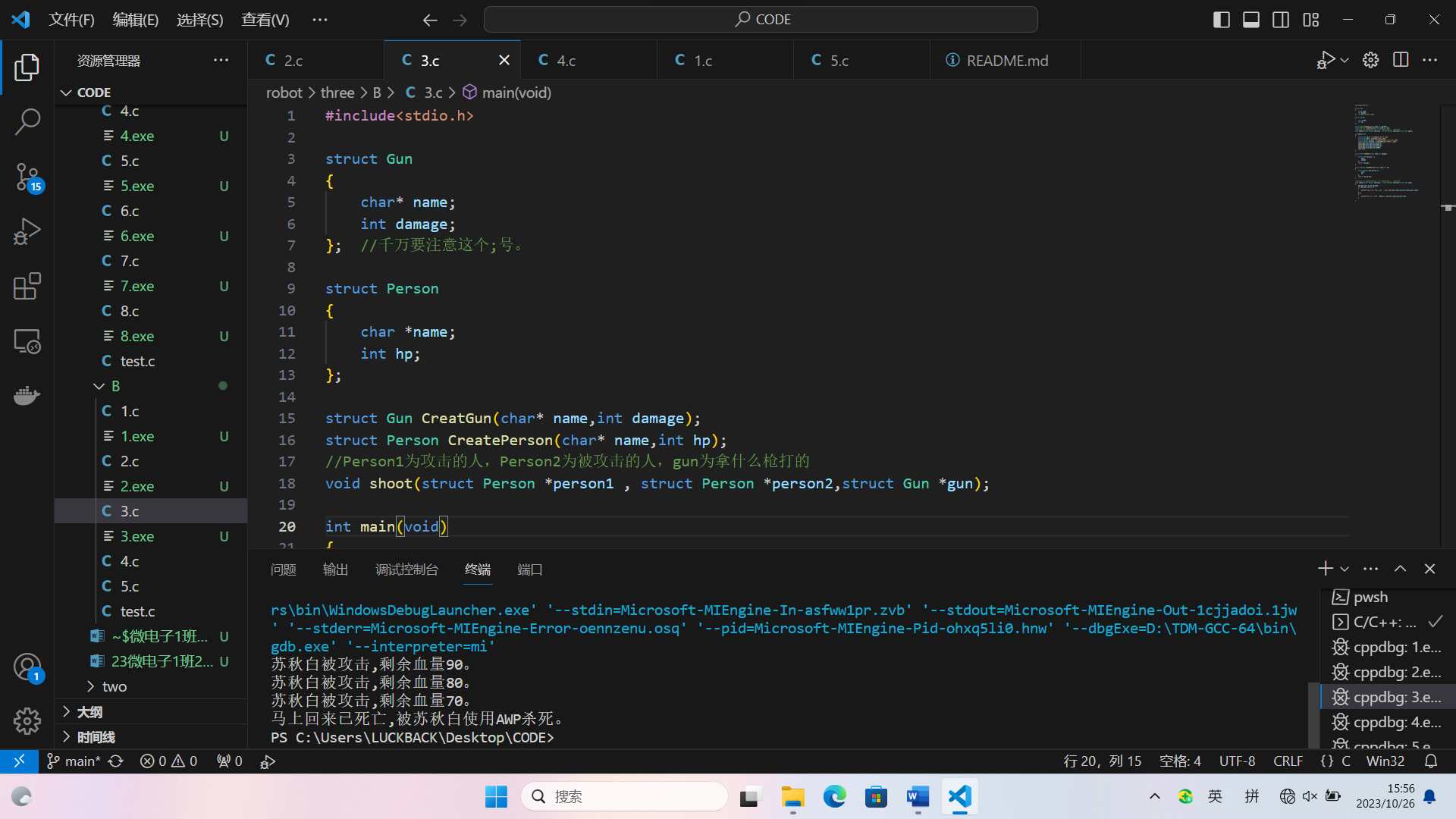
    else

    {

        printf("%s被攻击,剩余血量%d。\n",person2->name,person2->hp);

    }

}



第4题：

#include<stdio.h>

//结构体能够提高程序的可读性，知道每个变量的意义

struct Gun

{

    char\* name;

    int damage;

};  //千万要注意这个;号。

struct Person

{

    char \*name;

    int hp;

    int camp;//1表示警察，0表示匪徒

};

struct Gun CreatGun(char\* name,int damage);

struct Person CreatePerson(char\* name,int hp,int camp);

//Person1为攻击的人，Person2为被攻击的人，gun为拿什么枪打的

void shoot(struct Person \*person1 , struct Person \*person2,struct Gun \*gun);

//让他们肆意杀戮吧！！！

int main(void)

{

    struct Gun AK\_47 = CreatGun("AK-47",10);

    struct Gun AWP = CreatGun("AWP",100);

    struct Person person1 = CreatePerson("马上回来",100,1);

    struct Person person2 = CreatePerson("苏秋白",100,0);

    struct Person person3 = CreatePerson("Jrrr",100,1);

    shoot(&person1,&person2,&AK\_47);

    //shoot(&person3,&person1,&AWP);

    shoot(&person3,&person2,&AWP);

    shoot(&person1,&person3,&AK\_47);

    return 0;

}

//创建枪

struct Gun CreatGun(char\* name,int damage)

{

    struct Gun the\_gun = {

        name,

        damage

    };

    return the\_gun;

}

//创建人

struct Person CreatePerson(char\* name,int hp,int camp)

{

    struct Person the\_person ={

        name,

        hp,

        camp

    };

    return the\_person;

}

//Person1为攻击的人，Person2为被攻击的人，gun为拿什么枪打的

void shoot(struct Person \*person1 , struct Person \*person2,struct Gun \*gun)

{

    person2->hp -= gun->damage;

    //判断攻击的人是否死亡

    if (person1->hp<0)

    {

        printf("%s,你已经死了。",person1->name);

    }

    else

    {

        //查看是否是同一阵营

        if (person1->camp == person2->camp)

        {

            printf("%s,请你不要伤害队友。\n",person1->name);

        }

        else

        {

            //判断是否死亡

            if (person2->hp <= 0)

            {

                printf("%s已死亡,被%s使用%s杀死。\n",person2->name,person1->name,gun->name);

            }

            else

            {

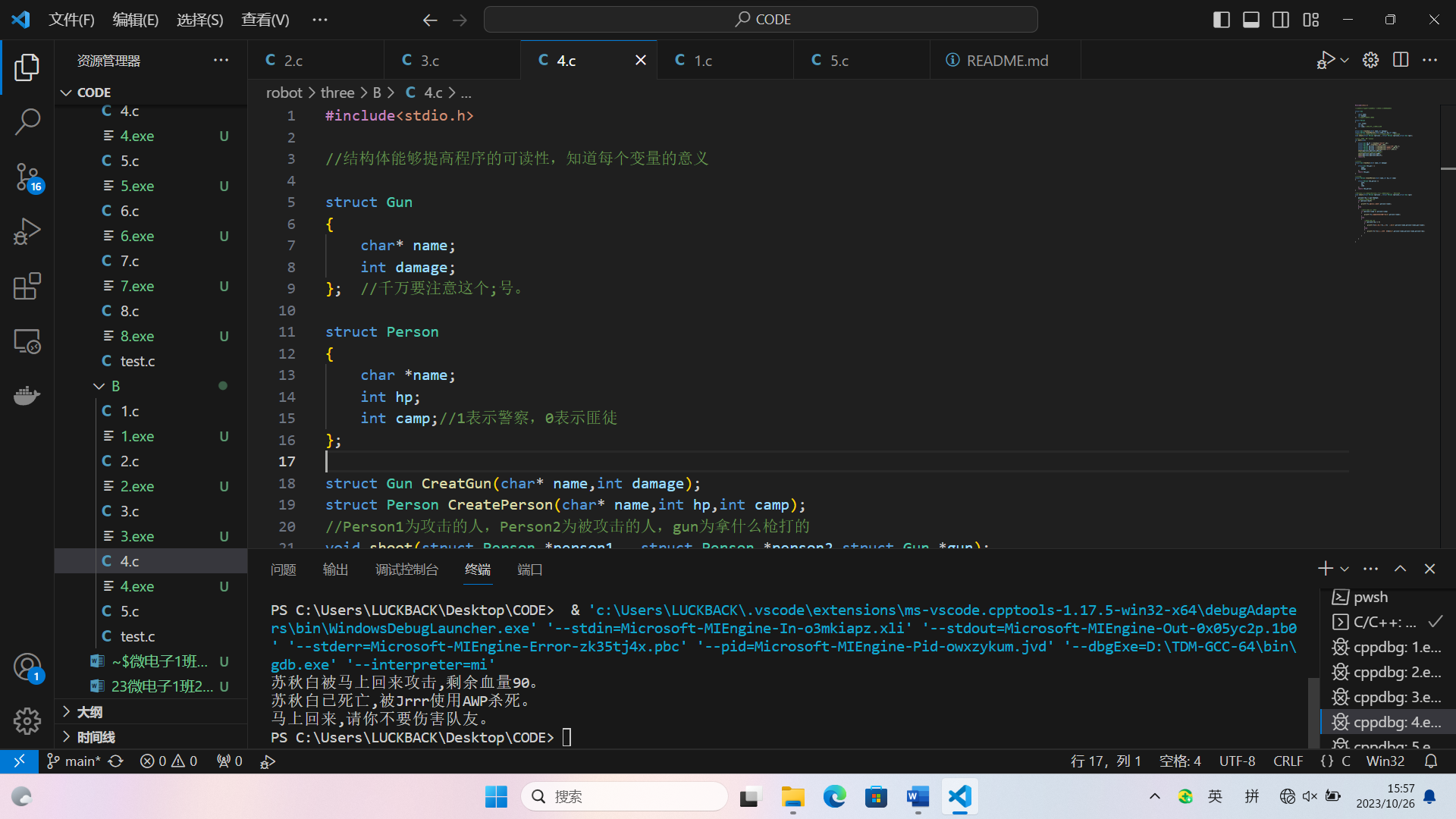
                printf("%s被%s攻击,剩余血量%d。\n",person2->name,person1->name,person2->hp);

            }

        }

    }

}



第5题：

不太懂题目的意思

#include<stdio.h>

#include<math.h>

//创建一个Point类型？

typedef struct point

{

    int x;

    int y;

    double distance;

} Point;

void MoveTo(Point\* pt, int x, int y);

double Distance(Point\* pt);

int main()

{

    Point the\_point = {0,0};

    MoveTo(&the\_point,1,1);

    printf("%lf",Distance(&the\_point));

}

//移动并计算距离

void MoveTo(Point\* pt, int x, int y)

{

    pt->distance = sqrt(pow((pt->x - x),2) + pow((pt->y - y),2));

    pt->x = x;

    pt->y = y;

}

//我不会弄，直接返回距离

double Distance(Point\* pt)

{

    return pt->distance;

}

