**定义复数及运算：**

**代码**：

/\* Complex.h \*/

#include <stdio.h>

/\*结构定义\*/

typedef struct {

float realpart;

float imagpart;

}Complex;

/\*基本操作的函数原型说明\*/

void Assign(Complex\* Z, float Realval, float Imagval);

//构造复数Z，其实部和虚部分别被赋以参数Realval和Imagval的值

float GetReal(Complex\*Z); //返回复数Z的实部值

float GetImag(Complex\*Z); //返回复数Z的虚部值

void Add(Complex z1, Complex z2, Complex\* sum); //返回加和结果sum

/\* Complex.c \*/

/\*基本操作的实现\*/

#include "Complex.h"

void Assign(Complex\* Z, float Realval, float Imagval) {

Z->imagpart = Imagval;

Z->realpart = Realval;

}

float GetReal(Complex \*Z) {

return (\*Z).realpart;

}

float GetImag(Complex \*Z) {

return (\*Z).imagpart;

}

void Add(Complex z1, Complex z2, Complex\* sum) {

sum->realpart = z1.realpart + z2.realpart;

sum->imagpart = z1.imagpart + z2.imagpart;

}

#include "Complex.h"

int main()

{

Complex com\_1, com\_2, sum;

Assign(&com\_1, 1.2, 2.3);

Assign(&com\_2, 2.1, 3.2);

Add(com\_1, com\_2, &sum);

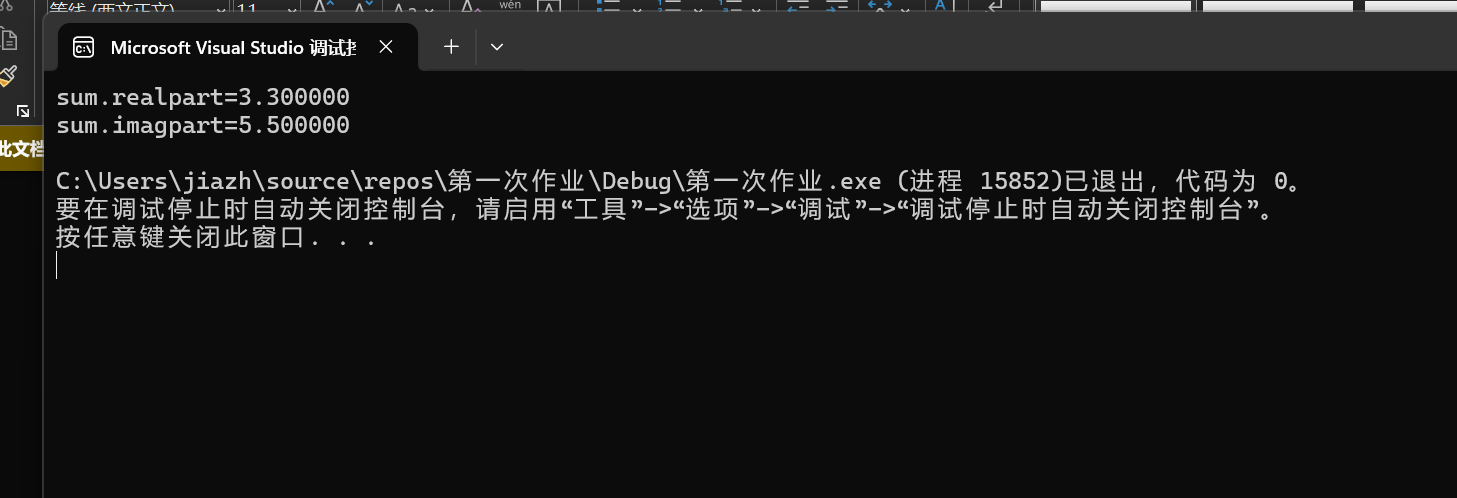
printf("sum.realpart=%f\nsum.imagpart=%f\n", sum.realpart, sum

.imagpart);

return 0;

}

**结果：**



**时间复杂度的探索:**

**代码：**

#include <stdio.h>

int countones(int N){

int ones = 0;

while (N > 0) {

ones += (1 & N);

N >>= 1;

}

return ones;

}

int main() {

int a=countones(150);

printf("%d", a);

return 0;

}

**结果：**