第一次实验

1.1 #include<iostream>

using namespace std;

int main()

{

int k = 1;

int i = k + 1;

cout << i++ << endl;

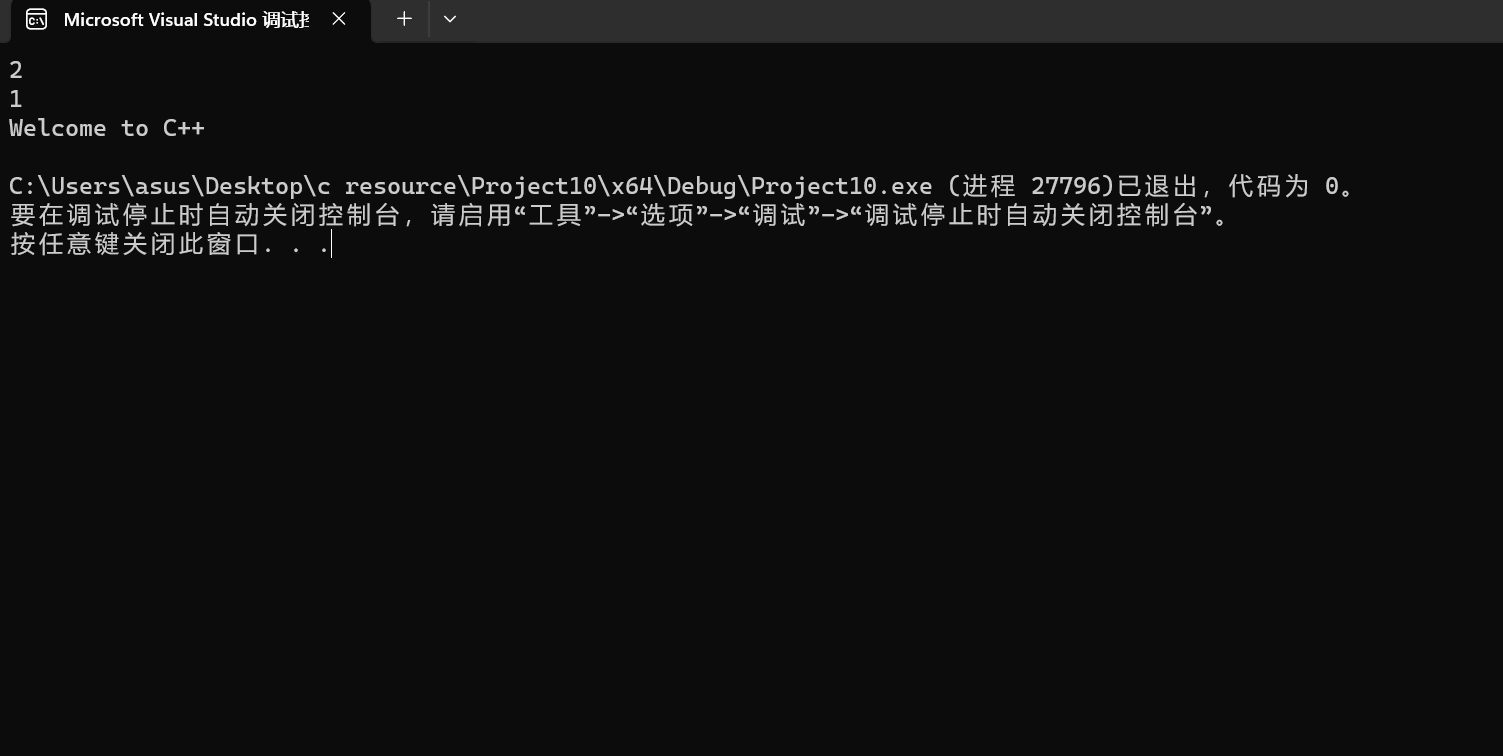
i=1;

cout << i++ << endl;

cout << "Welcome to C++" << endl;

return 0;

}



1.2 #include<iostream>

using namespace std;

int main()

{

const float pi = 3.14;

float r, h;

cout << "请输入圆锥底半径：" << endl;

cin >> r;

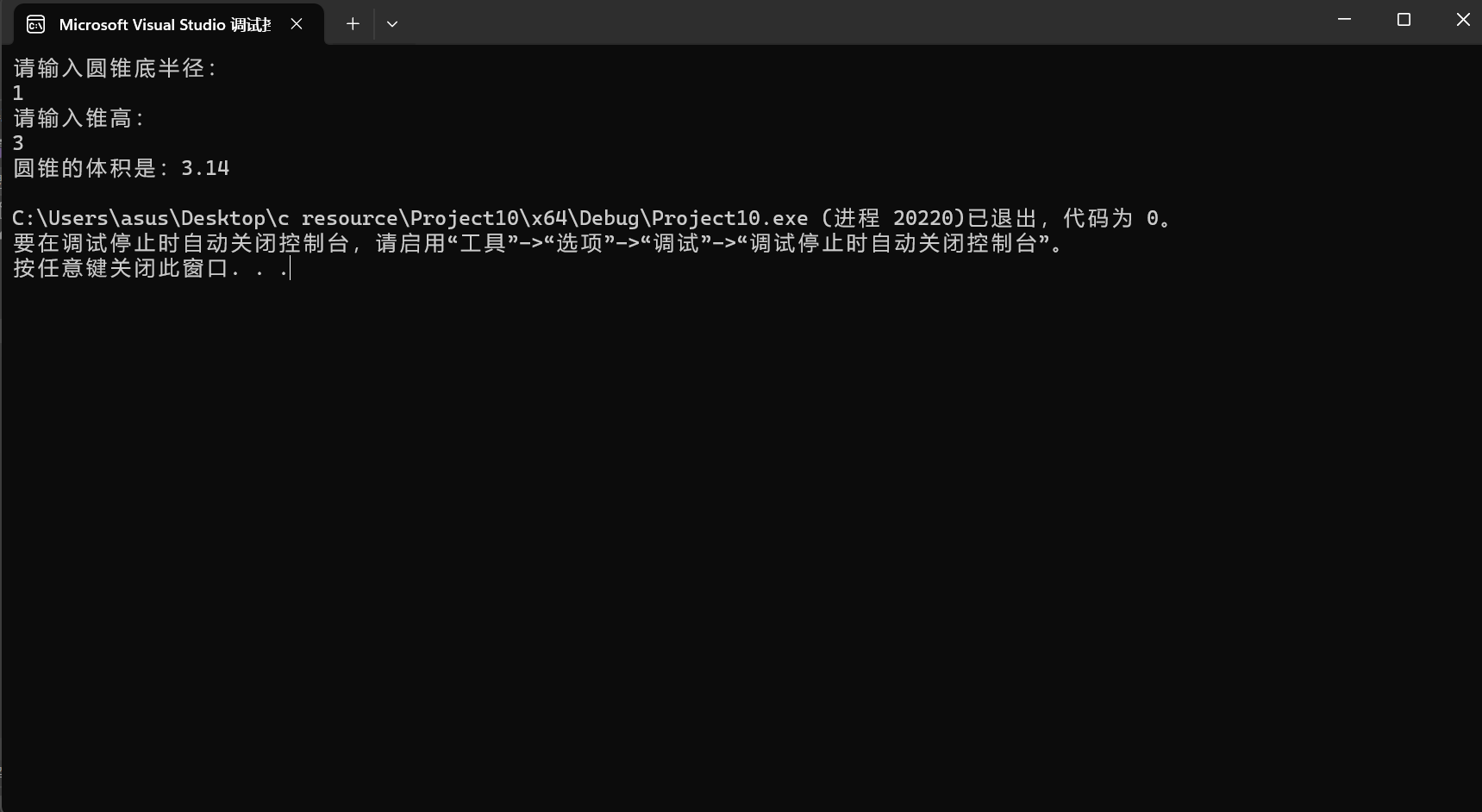
cout << "请输入锥高：" << endl;

cin >> h;

cout << "圆锥的体积是：" << pi \* h \* r \* r / 3 << endl;

return 0;

}



1.3 #include<iostream>

using namespace std;

int main()

{

cout << "char length:" << sizeof(char) << endl;

cout << "bool length:" << sizeof(bool) << endl;

cout << "unsigned char length:" << sizeof(unsigned char) << endl;

cout << "wchar\_t length:" << sizeof(wchar\_t) << endl;

cout << "int length:" << sizeof(int) << endl;

cout << "signed length:" << sizeof(signed) << endl;

cout << "unsigned length:" << sizeof(unsigned) << endl;

cout << "short length:" << sizeof(short) << endl;

cout << "long length:" << sizeof(long) << endl;

cout << "unsigned short length:" << sizeof(unsigned short) << endl;

cout << "unsigned long length:" << sizeof(unsigned long) << endl;

cout << "float length:" << sizeof(float) << endl;

cout << "double length:" << sizeof(double) << endl;

cout << "long double length:" << sizeof(long double) << endl;

return 0;

}



1.4 #include <iostream>

#include <iomanip>

using namespace std;

int main()

{

unsigned int testUnint = 65534;//oxfffe

cout << "output in unsigned int 1 type:"<< testUnint<< endl;//<<oct;

cout << "output in char type:!" << static\_cast<char>(testUnint) << endl;

cout << "output in short type:" << static\_cast<short>(testUnint) << endl;//为什么结果为-2?:

cout << "output in int type:" << static\_cast<int> (testUnint) << endl;

cout << "output in double type:" << static\_cast<double>(testUnint) << endl;

cout << "output in double type:" << setprecision(4) << static\_cast<double>(testUnint) << endl;

cout << "output in Hex unsigned int type:" << hex << testUnint << endl; //16进制输出

cout << "output in Oct unsigned int type:" << oct << testUnint << endl;

int a ;

cout << "input a:" << endl;

cin >> a;

cout << "output a:" << a << endl;

system("pause");

return 0;

}



1.5 #include<iostream>

#include<iomanip>

using namespace std;

int main()

{

float Fah, Cel;

cout << "请输入华氏温度：" << endl;

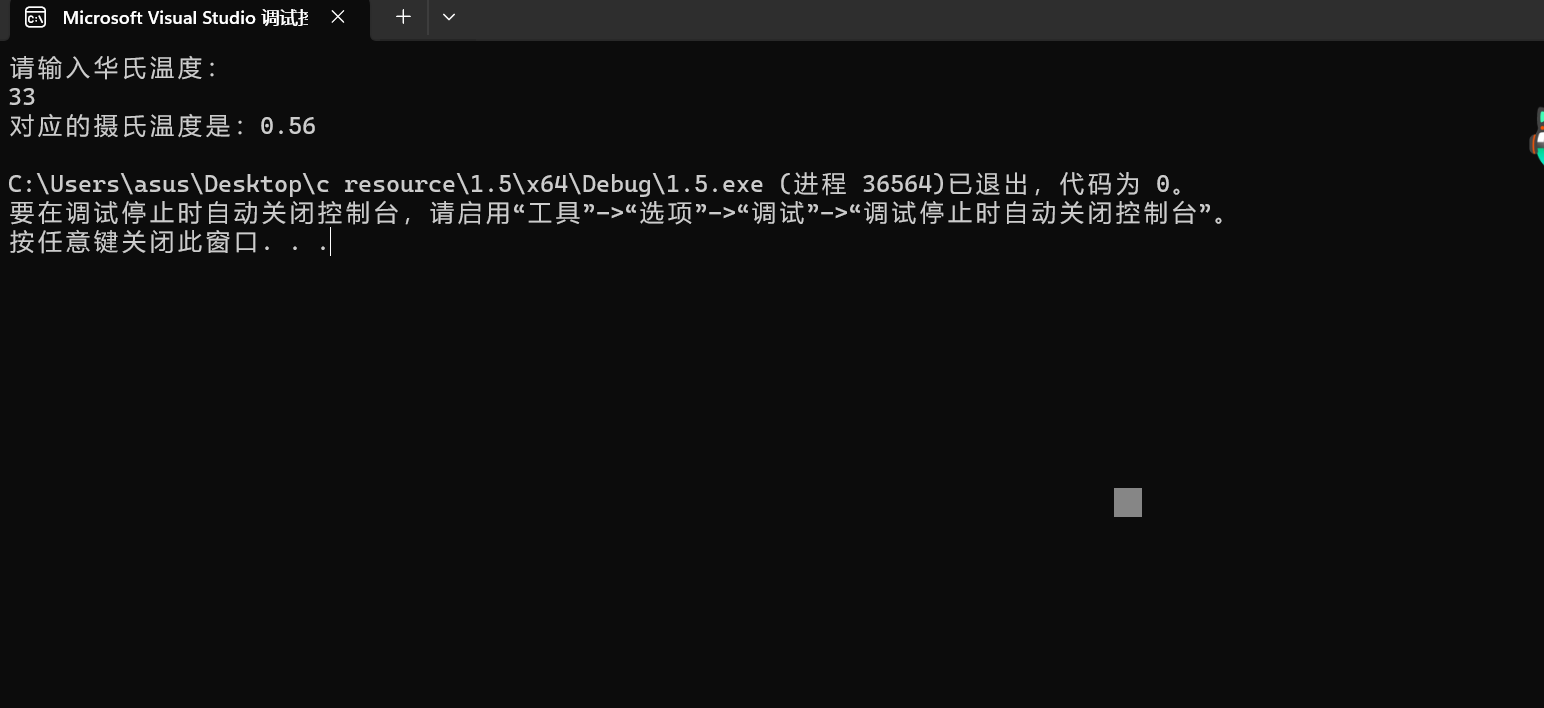
cin >> Fah;

Cel = (Fah - 32) / 1.8;

cout << "对应的摄氏温度是：" << setprecision(2) << Cel << endl;

return 0;

}



问题：1.4和1.5中使用setprecision时容易忘记预处理添加include<iomanip>；后续学习会加强对头文件预处理的重视。

第二次实验

2.1#include<iostream>

using namespace std;

int main()

{

cout << "请输入字符：" << endl;

char i;

cin >> i;

if (i >= 65 && i <= 90) {

i = i + 32;

cout << i << endl;

}

else if (i >= 97 && i <= 122) {

i = i - 32;

cout << i << endl;

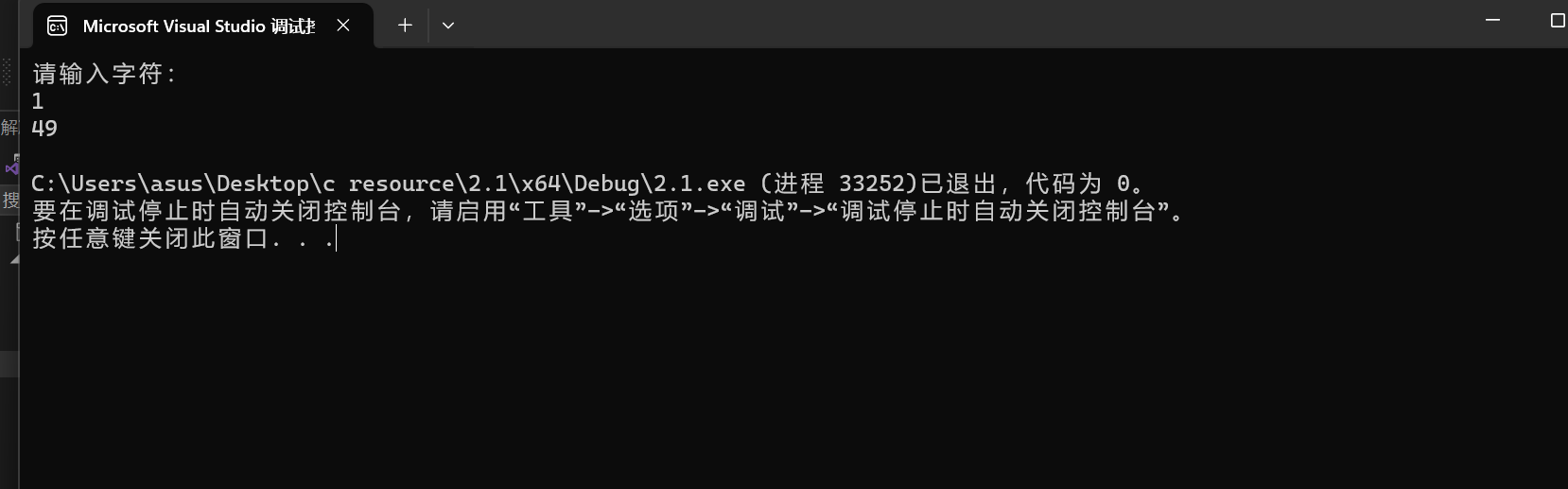
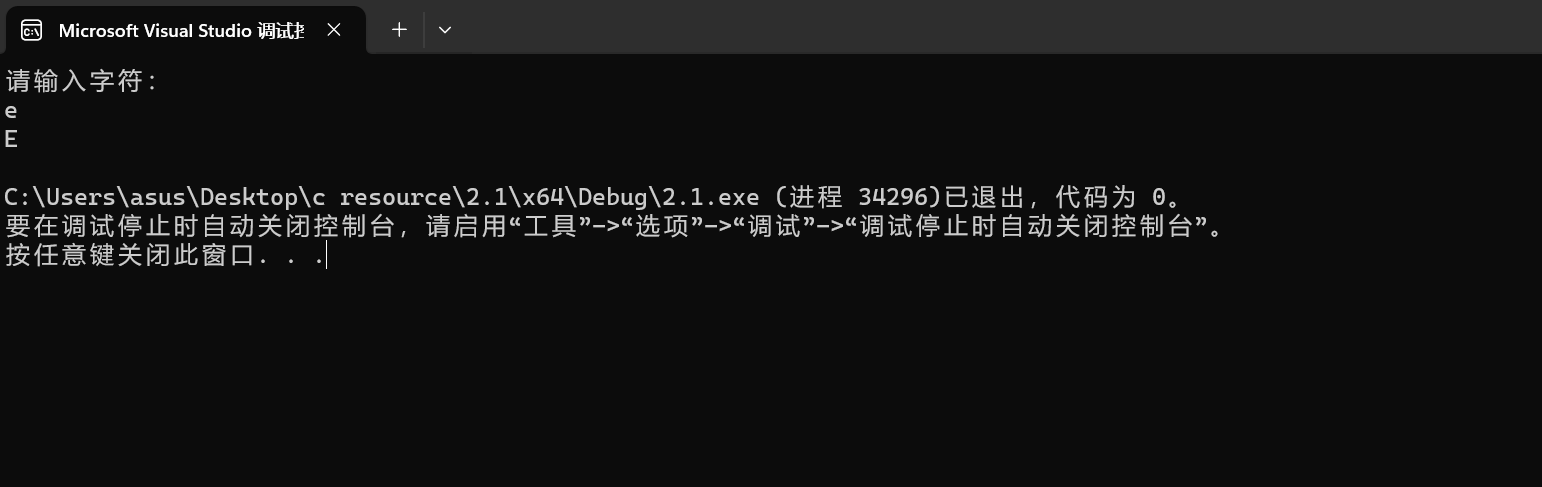
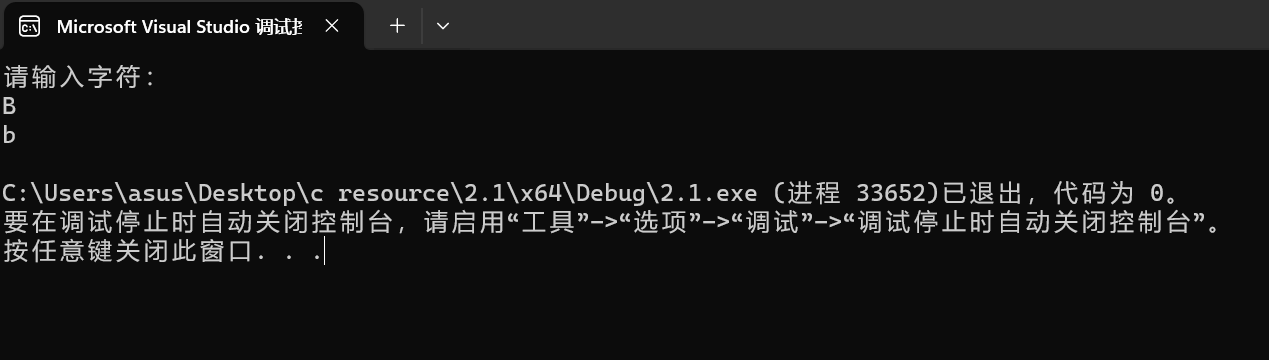
}

else {

cout << (int)i << endl;

}

}



2.2#include<iostream>

using namespace std;

int main()

{

float x, y;

cout << "请输入x的值：" << endl;

cin >> x;

if (x > 0 && x < 1) {

y = 3 - 2 \* x;

cout << "y的值是：" << y << endl;

}

else if (x >= 1 && x < 5) {

y = 2 / (4 \* x) + 1;

cout << "y的值是：" << y << endl;

}

else if (x >= 5 && x < 10) {

y = x \* x;

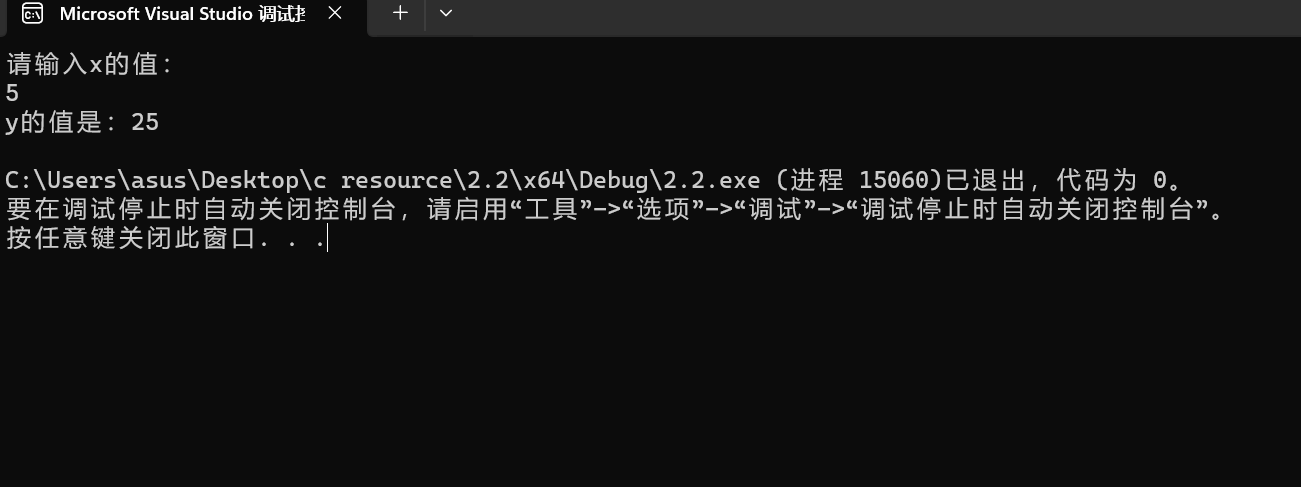
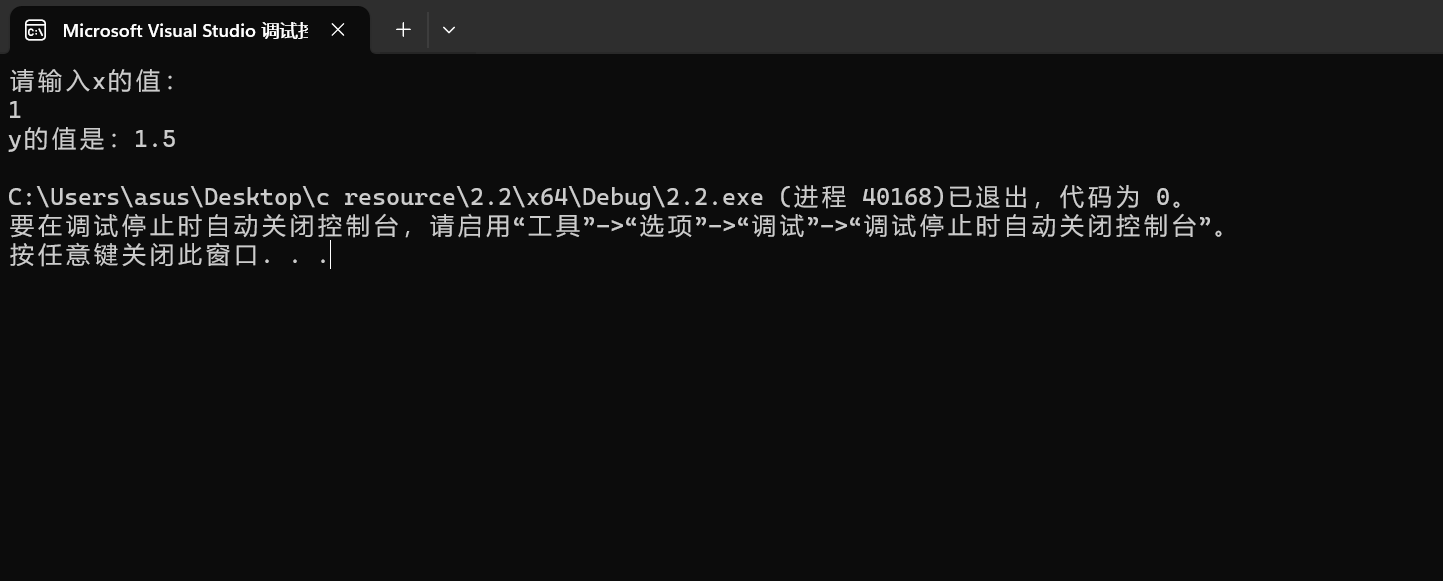
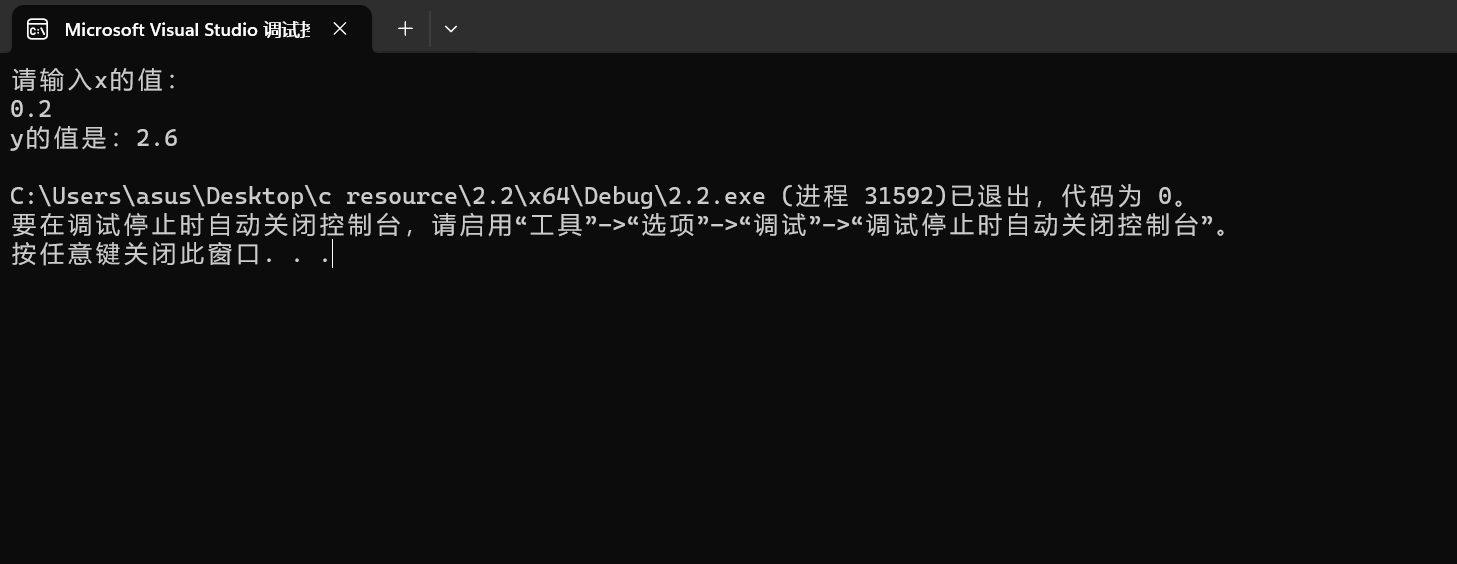
cout << "y的值是：" << y << endl;

}

else

cout << "错误！请输入正确范围内的x的值！" << endl;

}



2.3#include<iostream>

using namespace std;

int main()

{

float a, b, c, gir;

cout << "请输入三角形三边的长：" << endl;

cin >> a>> b>> c;

float t;

t = (a > b ? (a > c ? a : c) : (b > c ? b : c));

gir = a + b + c;

if (gir - t <= t) {

cout << "这三边不能构成三角形！" << endl;

}

else {

cout << "这个三角形的周长是：" << gir << endl;

if (a == b || b == c || a == c) {

cout << "这个三角形是等腰三角形！" << endl;

}

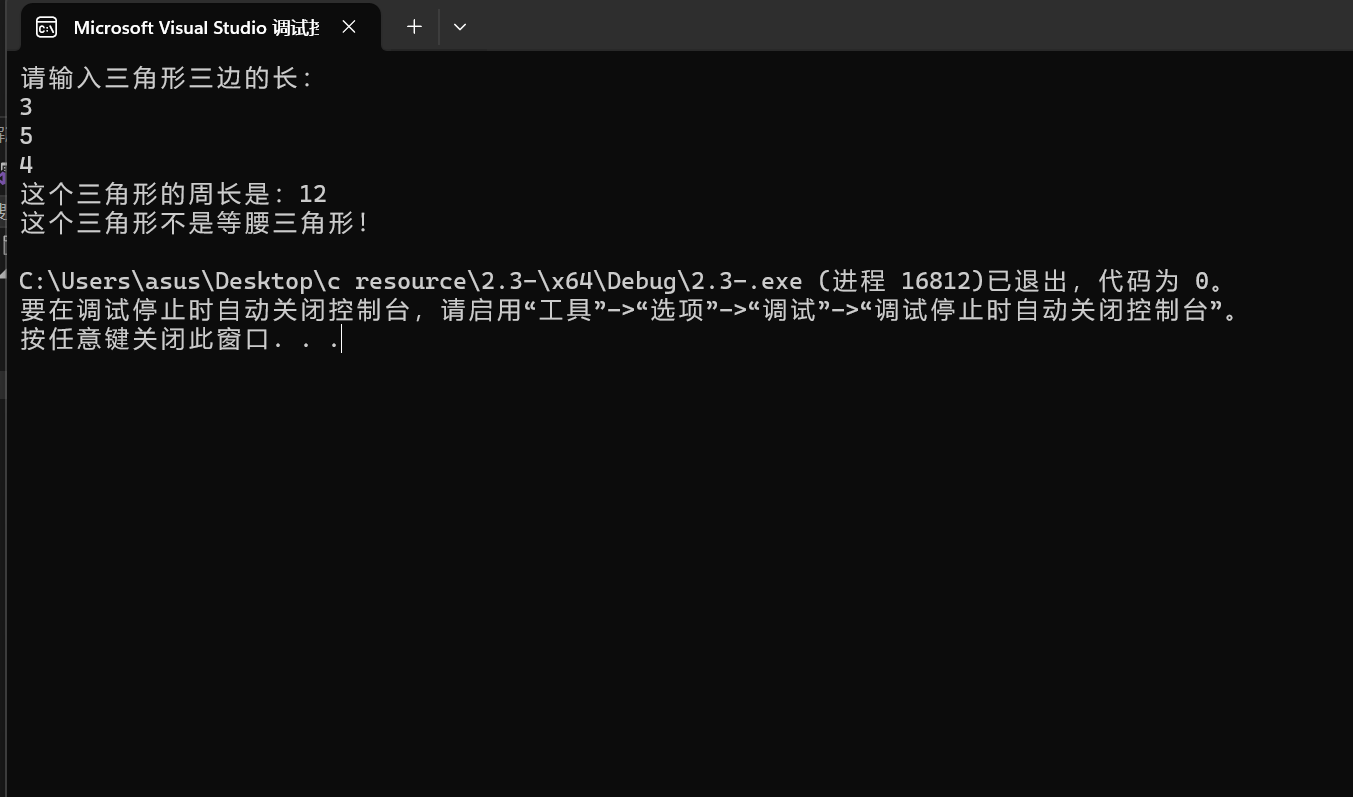
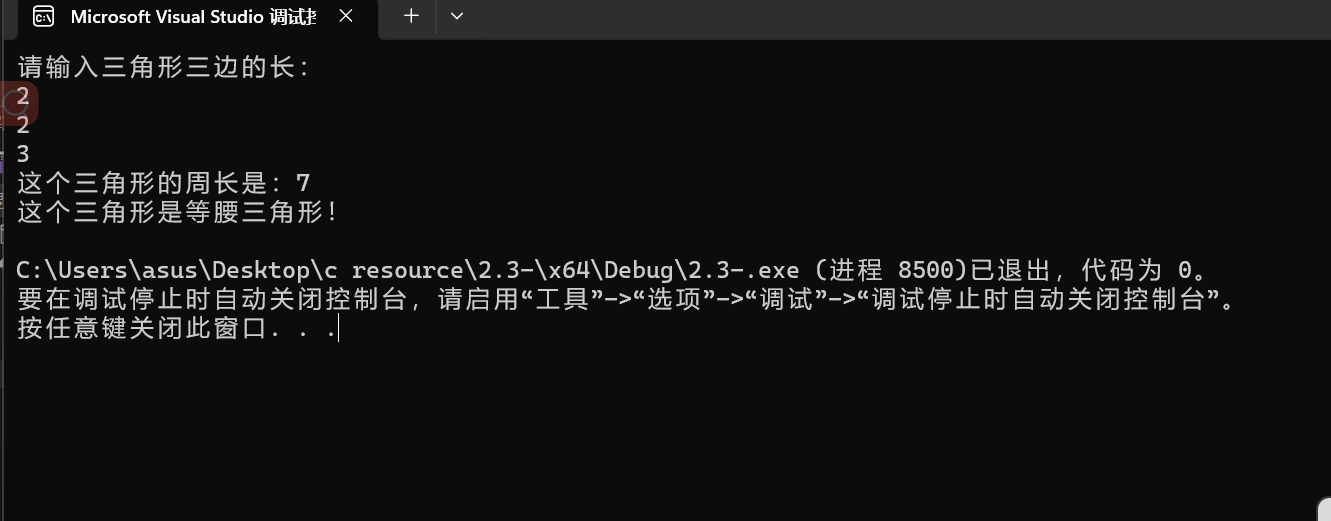
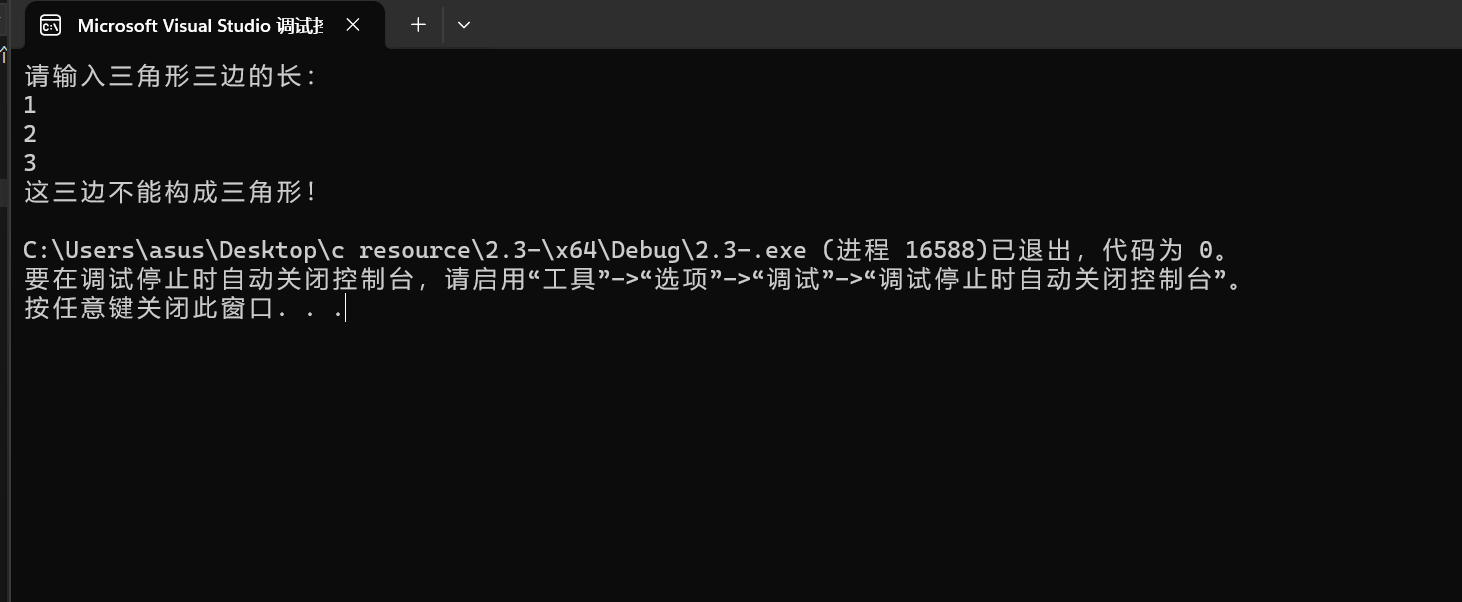
else {

cout << "这个三角形不是等腰三角形！" << endl;

}

}

}



2.4#include<iostream>

using namespace std;

int main()

{

int x, y;

char i;

cout << "请输入第一个数：" << endl;

cin >> x;

cout << "请输入运算符：" << endl;

cin >> i;

cout << "请输入第二个数：" << endl;

cin >> y;

switch (i) {

case'+': {

cout << x + y << endl;

break;

}

case'-': {

cout << x - y << endl;

break;

}

case'\*': {

cout << x \* y << endl;

break;

}

case'/': {

if (y == 0) {

cout << "除数不可以为0！" << endl;

break;

}

else {

cout << x / y << endl;

break;

}

}

case'%': {

cout << x % y << endl;

break;

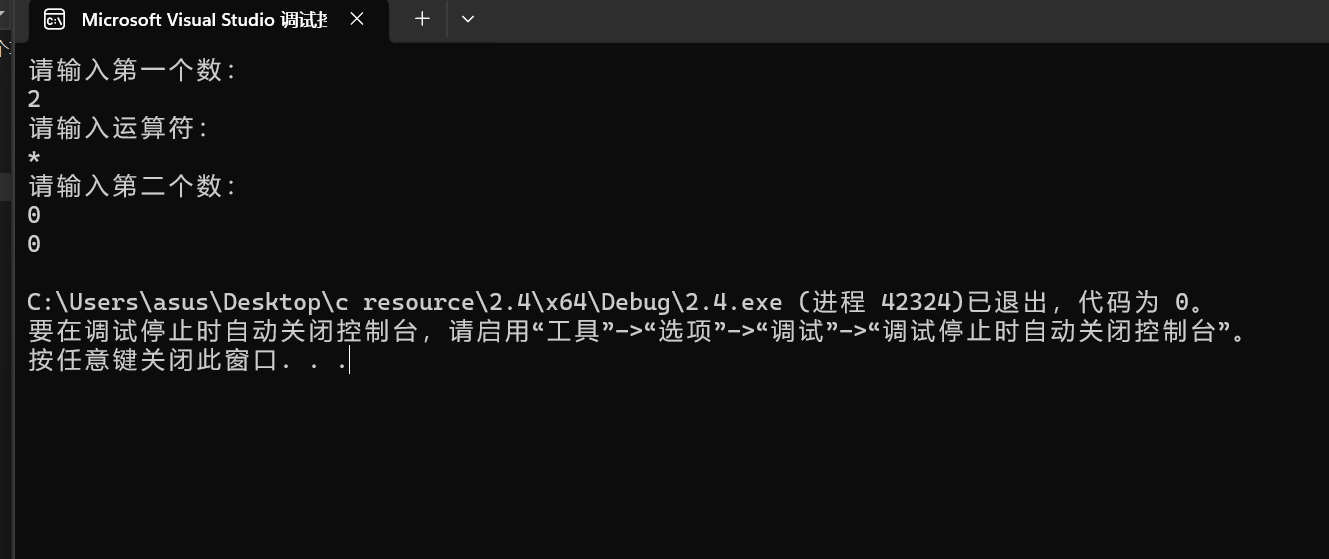
default:cout << "请输入正确的运算符！" << endl;

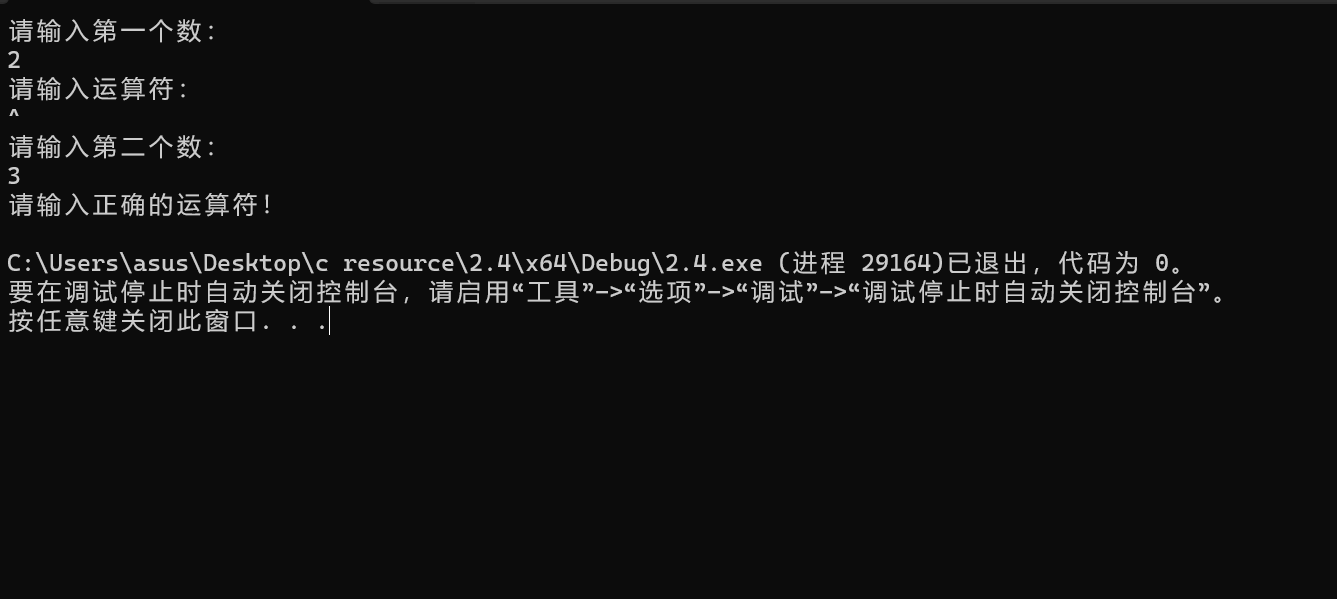
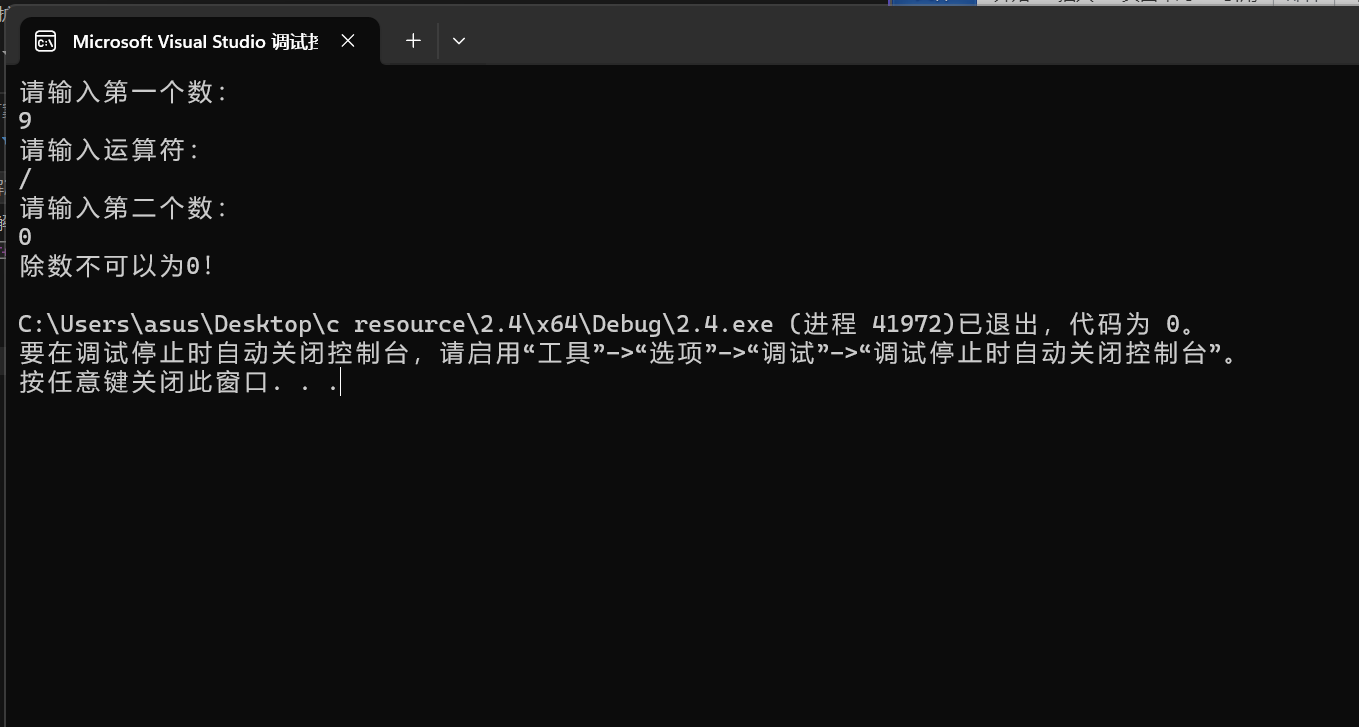
}

return 0;

}

}





2.5#include<iostream>

using namespace std;

int main()

{

char ch;

cout << "请输入一行字符：" << endl;

int letter=0, space=0, digit=0, others=0;

while ((ch=getchar())!= '\n')

{

if ((ch >= 'A' && ch <= 'Z') || (ch >= 'a' && ch <= 'z'))

letter++;

else if (ch >='0' && ch <= '9')

digit++;

else if (ch == ' ')

space++;

else

others++;

}

cout << "letternum is " << letter << endl;

cout << "spacenum is " << space << endl;

cout << "digitnum is " << digit << endl;

cout << "othersnum is " << others << endl;

}



2.6#include<iostream>

using namespace std;

int main()

{

int x, y;

cout << "请输入两个正整数：" << endl;

cin >> x >> y;

for (int i = (x < y ? x : y);;i--) {

if (x % i == 0 && y % i == 0) {

cout << "最大公约数是：" << i << endl;

break;

}

}

for (int j = (x < y ? y : x);;j++) {

if (j % x == 0 && j % y == 0) {

cout << "最小公倍数是：" << j << endl;

break;

}

}

}



2.7#include<iostream>

using namespace std;

int main()

{

for (int i = 0; i <= 5;i++) {

for (int j = 0;j < i;j++) {

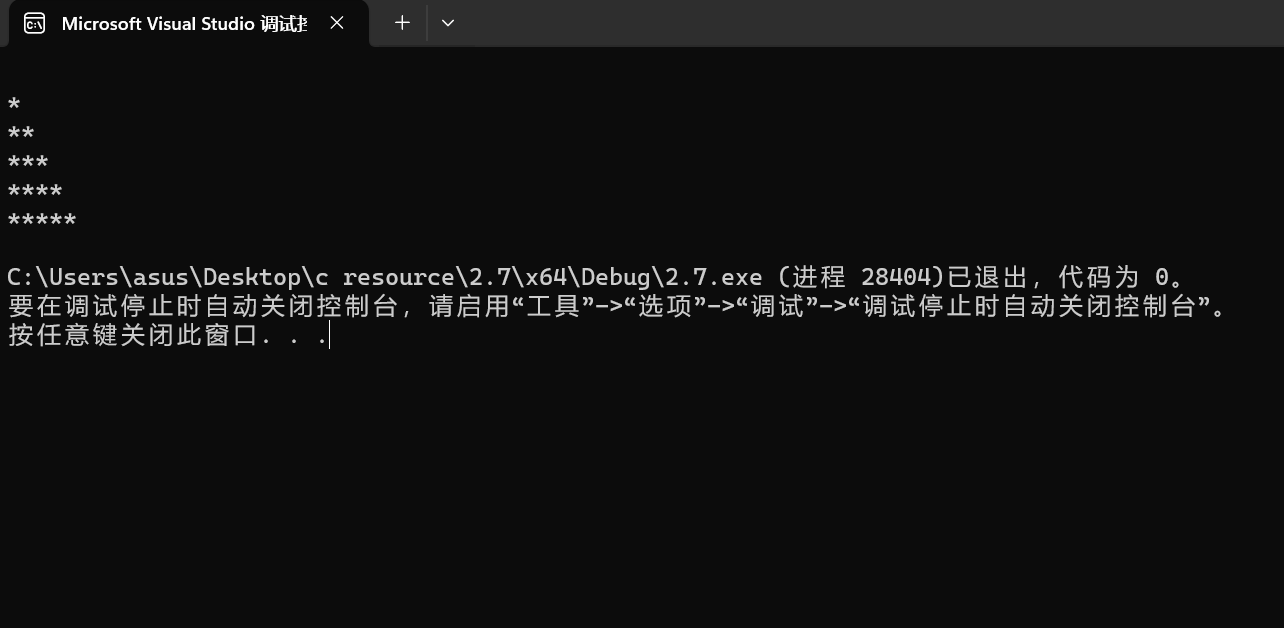
cout << '\*';

}

cout << endl;

}

}



2.8#include<iostream>

using namespace std;

int main()

{

double a;

cout << "请输入a：" << endl;

cin >> a;

if (a < 0) {

cout << "请输入一个非负实数！" << endl;

}

else {

double x = a;

double old = a + 1;

while (fabs(x - old) > 10e-5) {

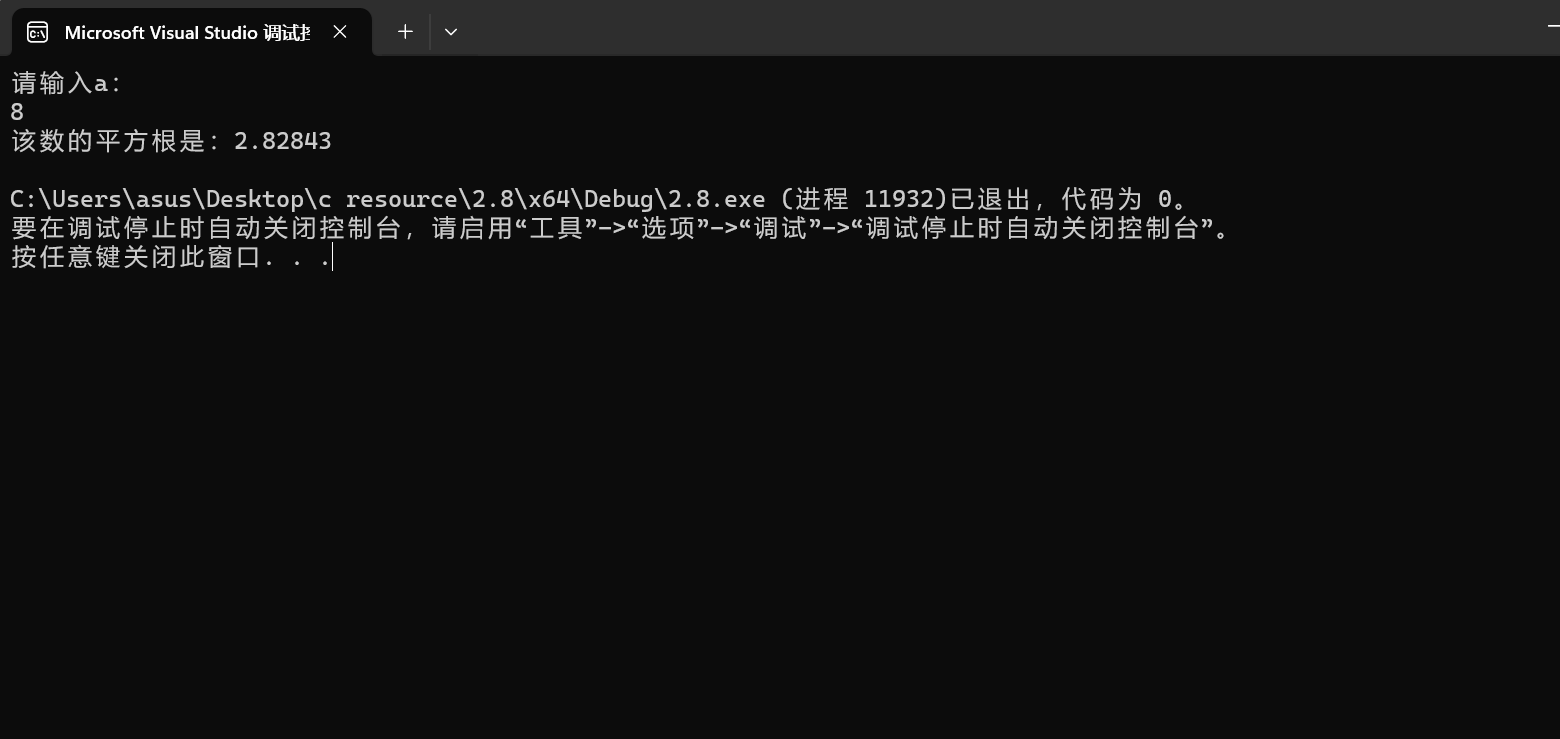
old = x;

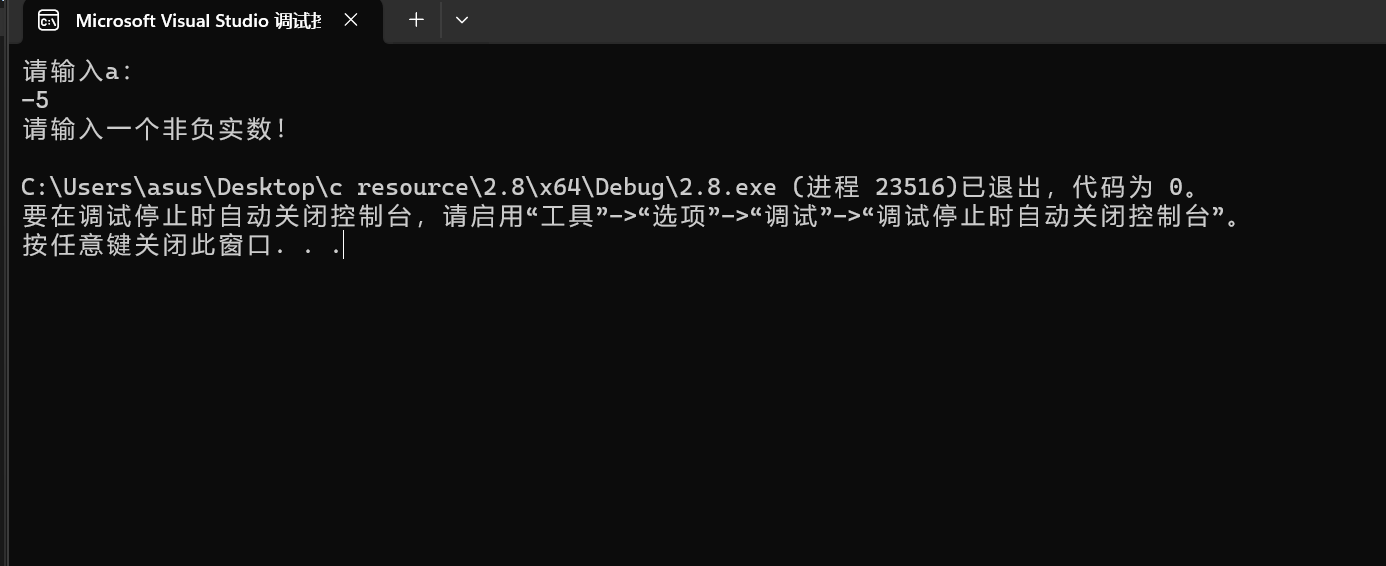
x = (old + a / old) / 2;

}

cout << "该数的平方根是：" << x << endl;

}





2.9#include<iostream>

using namespace std;

int main() {

int each = 2;

int sum = 2;

int i = 1;

for (;sum<=100;) {

each \*= 2;

sum = sum + each;

i++;

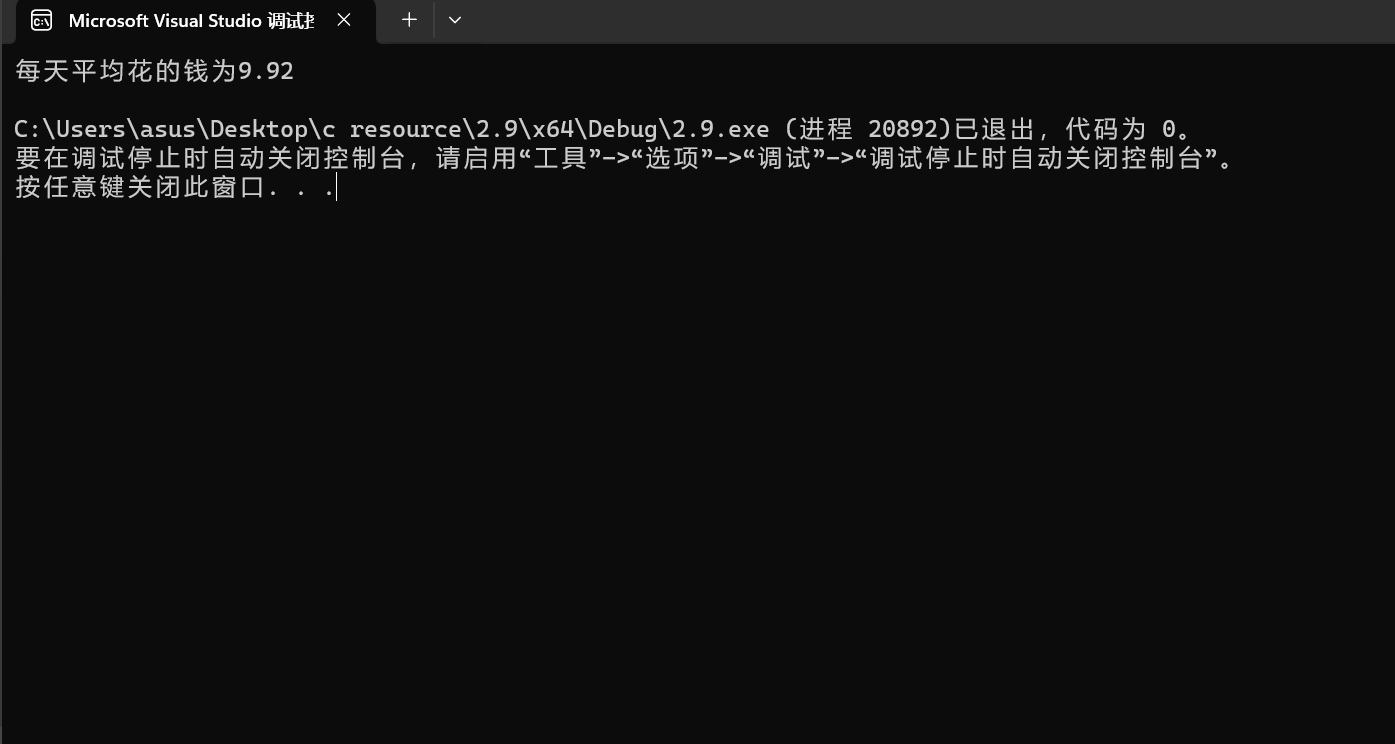
}

float perday;

perday = 0.8 \*(sum-each) /(--i);

cout << "每天平均花的钱为" << perday << endl;

}



问题：对循环体的使用还不够熟练，例如for循环中容易对循环条件审视不清，且在1.8的迭代中容易导致代码顺序混乱引起程序结果错误。嵌套循环时也容易思维混乱。后续学习中会加以强化熟练。