**6.21**

#include <iostream>

using namespace std;

bool isEven(int);

int main()

{

int num;

while (cin >> num) {

//输入检查

if (num < 0) {

cout << "Illegal input!" << endl;

continue;

}

//输出

if (isEven(num))

cout << num << " is an even number." << endl;

else

cout << num << " is an odd number." << endl;

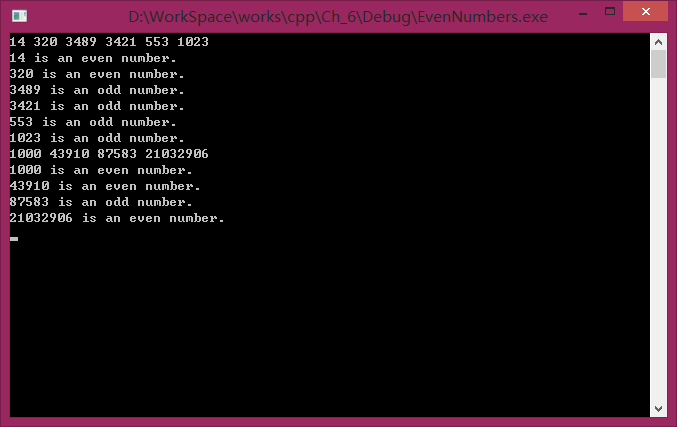
}

}

bool isEven(int i) {

return i % 2 == 0;

}



**6.34**

#include <iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

enum Status {HIGH, LOW, CORRECT, CONTINUE, EXIT};

/\*

\* HIGH: 所猜数太高

\* LOW: 所猜数太低

\* CORRECT: 猜测正确

\* CONTINUE: 继续游戏标记

\* EXIT: 退出游戏标记

\*/

Status check(int, int);

int main() {

int target;//目标数字

int input;//用户猜数

char sign;//y/n输入

Status currentStatus = CONTINUE;//当前游戏状态

while (cin) {

if (currentStatus == EXIT)//标记为EXIT时退出游戏

break;

srand(time(0));

target = 1 + rand() % 1000;//构造待猜数

cout << "I have a number between 1 to 1000.\nCan you guess my number?\nPlease type your first guess.\n";

while (currentStatus != CORRECT) {

cin >> input;

currentStatus = check(input, target);

switch (currentStatus) {

case HIGH:

cout << "Too high! Try again.\n";

break;

case LOW:

cout << "Too low! Try again.\n";

break;

case CORRECT:

cout << "Excellent! You guess the number.\nWould you want to play again? (y or n)\n";

break;

}

}

cin >> sign;//获取y/n输入

if (sign == 'y')

currentStatus = CONTINUE;

else if (sign == 'n')

currentStatus = EXIT;

else

/\*非法输入处理\*/

break;

}

}

Status check(int input, int target) {

if (input > target)

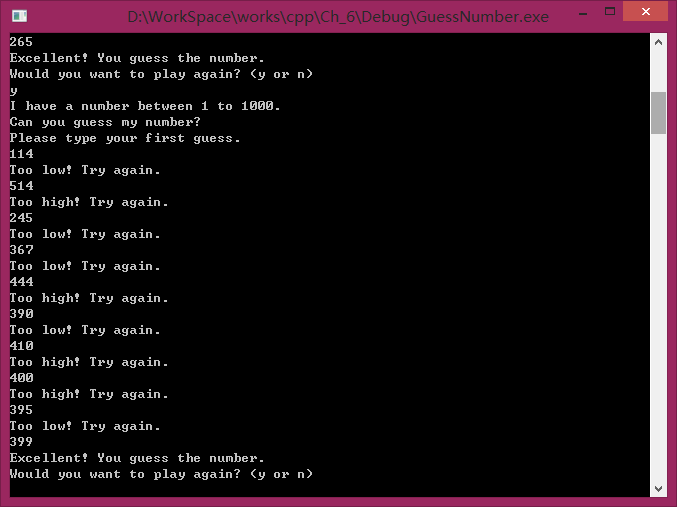
return HIGH;

else if (input < target)

return LOW;

return CORRECT;

}



**6.35**

#include <iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

enum Status {HIGH, LOW, CORRECT, CONTINUE, EXIT};

/\*

\*HIGH: 所猜数太高

\*LOW: 所猜数太低

\*CORRECT: 猜测正确

\*CONTINUE: 继续游戏标记

\*EXIT: 退出游戏标记

\*/

Status check(int, int);

int main() {

int target;//目标数字

int input;//用户猜数

char sign;//y/n输入

int times;//猜测次数

Status currentStatus = CONTINUE;//当前游戏状态

while (cin) {

times = 0;//初始化猜测次数

if (currentStatus == EXIT)//标记为EXIT时退出游戏

break;

srand(time(0));

target = 1 + rand() % 1000;//构造待猜数

cout << "I have a number between 1 to 1000.\nCan you guess my number?\nPlease type your first guess.\n";

while (currentStatus != CORRECT) {

cin >> input;

times++;

currentStatus = check(input, target);

switch (currentStatus) {

case HIGH:

cout << "Too high! Try again.\n";

break;

case LOW:

cout << "Too low! Try again.\n";

break;

case CORRECT:

cout << "Excellent! You guess the number.\n";

break;

}

}

//评价

cout << "\nYou guessed " << times << " times.\n";

if (times > 10)

cout << "You should be able to do better.\n";

else if (times < 10)

cout << "Either you know the secret or you got lucky!\n";

else

cout << "Ahah! You know the secret!\n";

cout << "Would you want to play again ? (y or n)\n";

cin >> sign;//获取y/n输入

if (sign == 'y')

currentStatus = CONTINUE;

else if (sign == 'n')

currentStatus = EXIT;

else

/\*非法输入处理\*/

break;

}

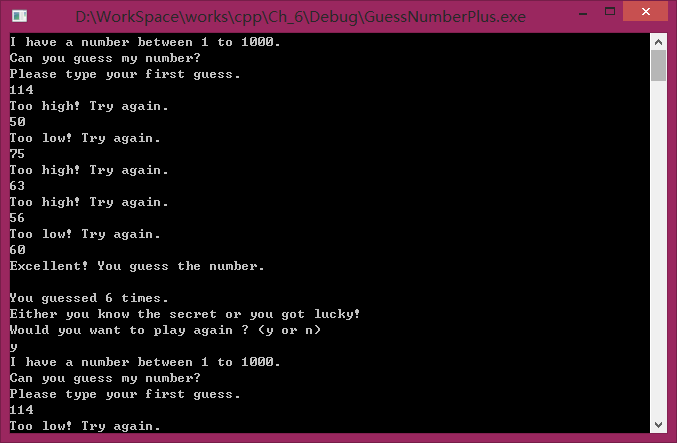
}

Status check(int input, int target) {

if (input > target)

return HIGH;

else if (input < target)

 return LOW;

return CORRECT;

}

**5.20 Follow-up**

**1.**

#include <iostream>

using namespace std;

int main() {

int s1,s2,s3,loopCounter = 0, triangleCounter = 0;

cout << "Side1\tSide2\tSide3\n";

for (s1 = 1; s1 <= 500; s1++)

for (s2 = s1; s2 <= 500; s2++)

for (s3 = s2; s3 <= 500; s3++) {

if (s1 \* s1 + s2 \* s2 == s3 \* s3) {

cout << s1 << "\t" << s2 << "\t" << s3 << endl;

triangleCounter++;

}

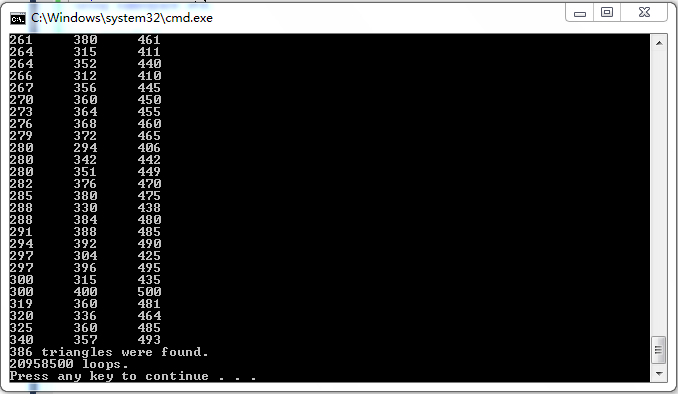
loopCounter++;

}

cout << triangleCounter << " triangles were found." << endl;

cout << loopCounter << " loops." << endl;

}



**2.**

**Innermost:**

#include <iostream>

using namespace std;

int main() {

int s1,s2,s3, loopCounter = 0, triangleCounter = 0;

cout << "Side1\tSide2\tSide3\n";

for (s1 = 1; s1 <= 500; s1++) {

for (s2 = s1; s2 <= 500; s2++) {

for (s3 = s2; s3 <= 500; s3++) {

loopCounter++;

if (s1 \* s1 + s2 \* s2 == s3 \* s3) {

cout << s1 << "\t" << s2 << "\t" << s3 << endl;

triangleCounter++;

if (triangleCounter >= 20)

break;

}

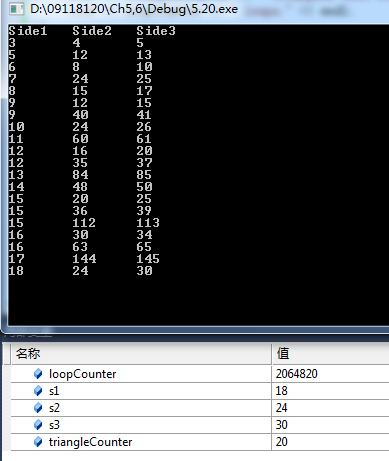
}

}

}

cout << triangleCounter << " triangles were found." << endl;

cout << loopCounter << " loops." << endl;

}

**Middle:**

#include <iostream>

using namespace std;

int main() {

int s1,s2,s3, loopCounter = 0, triangleCounter = 0;

cout << "Side1\tSide2\tSide3\n";

for (s1 = 1; s1 <= 500; s1++) {

for (s2 = s1; s2 <= 500; s2++) {

for (s3 = s2; s3 <= 500; s3++) {

loopCounter++;

if (s1 \* s1 + s2 \* s2 == s3 \* s3) {

cout << s1 << "\t" << s2 << "\t" << s3 << endl;

triangleCounter++;

}

}

if (triangleCounter >= 20)

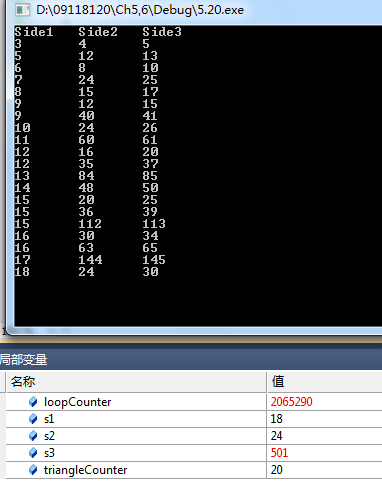
break;

}

}

cout << triangleCounter << " triangles were found." << endl;

cout << loopCounter << " loops." << endl;

}

**Outermost:**

#include <iostream>

using namespace std;

int main() {

int s1,s2,s3, loopCounter = 0, triangleCounter = 0;

cout << "Side1\tSide2\tSide3\n";

for (s1 = 1; s1 <= 500; s1++) {

for (s2 = s1; s2 <= 500; s2++) {

for (s3 = s2; s3 <= 500; s3++) {

loopCounter++;

if (s1 \* s1 + s2 \* s2 == s3 \* s3) {

cout << s1 << "\t" << s2 << "\t" << s3 << endl;

triangleCounter++;

}

}

}

if (triangleCounter >= 20)

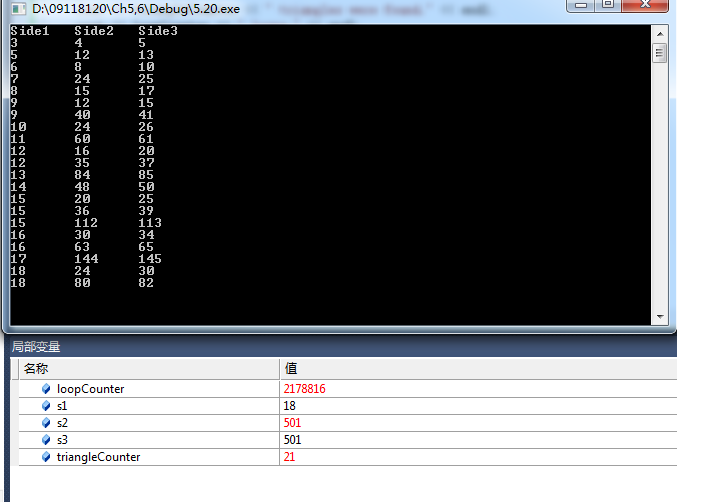
break;

}

cout << triangleCounter << " triangles were found." << endl;

cout << loopCounter << " loops." << endl;

}



**3.**

#include <iostream>

using namespace std;

int main() {

int s1,s2,s3, loopCounter = 0, triangleCounter = 0;

cout << "Side1\tSide2\tSide3\n";

for (s1 = 1; s1 <= 500; s1++) {

if (s1 == 8)

continue;

for (s2 = s1; s2 <= 500; s2++) {

for (s3 = s2; s3 <= 500; s3++) {

loopCounter++;

if (s1 \* s1 + s2 \* s2 == s3 \* s3) {

cout << s1 << "\t" << s2 << "\t" << s3 << endl;

triangleCounter++;

}

}

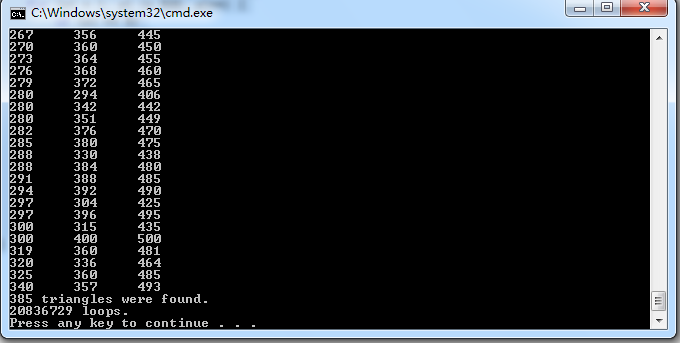
}

}

cout << triangleCounter << " triangles were found." << endl;

cout << loopCounter << " loops." << endl;

}



\*continue跳过当前循环的剩余语句但不退出循环。

**4.**

#include <iostream>

using namespace std;

int main() {

int s1,s2,s3, loopCounter = 0, triangleCounter = 0;

cout << "Side1\tSide2\tSide3\n";

for (s1 = 1; s1 <= 500; s1++) {

for (s2 = s1; s2 <= 500; s2++) {

for (s3 = s2; s3 <= 500; s3++) {

loopCounter++;

int a = short(s1 \* s1 + s2 \* s2);

if (short(s1 \* s1 + s2 \* s2) == short(s3 \* s3)) {

cout << s1 << "\t" << s2 << "\t" << s3 << endl;

triangleCounter++;

}

}

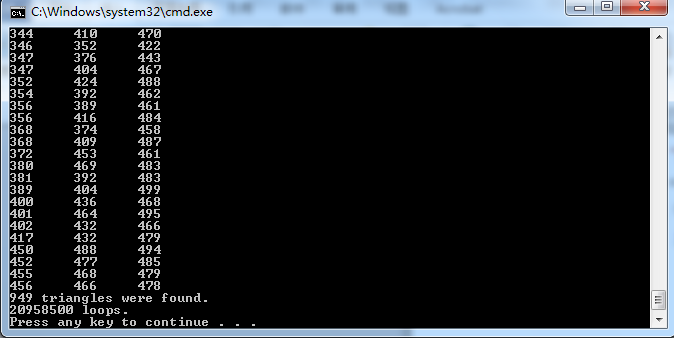
}

}

cout << triangleCounter << " triangles were found." << endl;

cout << loopCounter << " loops." << endl;

}



\*short会将超出范围的int截断导致出错。

**6.30 Follow-up**

**1.**

#include <iostream>

using namespace std;

int reverse(int);

int main() {

int num;

while (cin) {

cout << "Type a number: ";

cin >> num;

cout << reverse(reverse(num)) << endl;

}

}

int reverse(int num) {

int temp = 0;

while (num != 0) {

temp \*= 10;

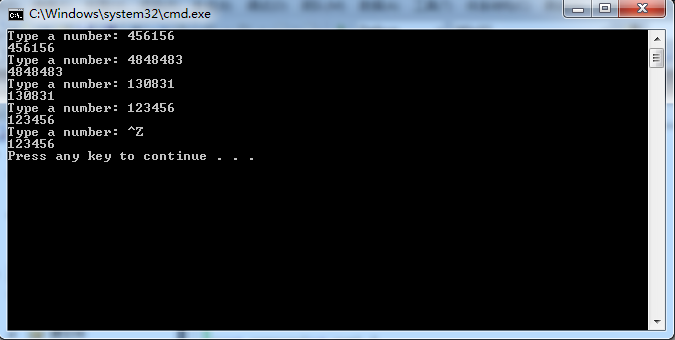
temp += num % 10;

num /= 10;

}

return temp;

}



**2.**

#include <iostream>

using namespace std;

void reverse(int&);

int main() {

int num;

while (cin) {

cout << "Type a number: ";

cin >> num;

reverse(num);

cout << num << endl;

}

}

void reverse(int& num) {

int temp = 0;

while (num != 0) {

temp \*= 10;

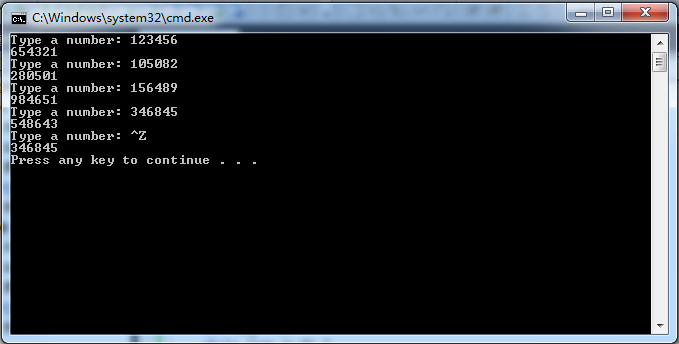
temp += num % 10;

num /= 10;

}

num = temp;

}



**6.31**

#include <iostream>

using namespace std;

int gcd(int,int);

int main() {

int a,b;

while (cin) {

cout << "Type 2 numbers: ";

cin >> a >> b;

cout << "The GCD is " << gcd(a,b)<< endl;

}

}

int gcd(int a, int b) {

int min = a;

int temp = 1;

if (a > b)

min = b;

for (int i = 1; i <= min; i++) {

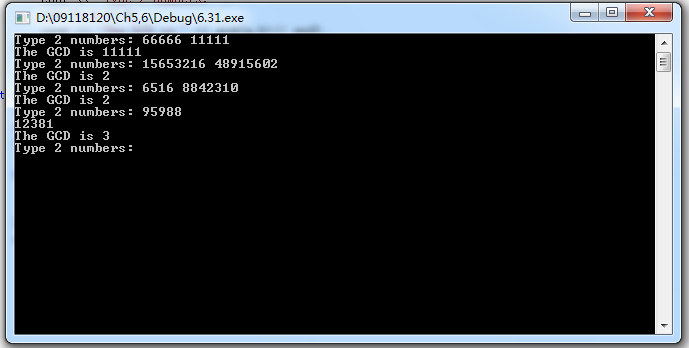
if (a % i == 0 && b % i == 0)

temp = i;

}

return temp;

}



**6.38 Follow-up**

#include <iostream>

using namespace std;

void tower(int,int,int,int);

static int counter = 0;

int main() {

tower(4,1,2,3);

system("pause");

}

void tower(int num, int start, int temp, int end) {

if (num == 1) {

counter++;

cout << counter << ": " << start << "-->" << end << endl;

} else {

tower(num - 1, start, end, temp);

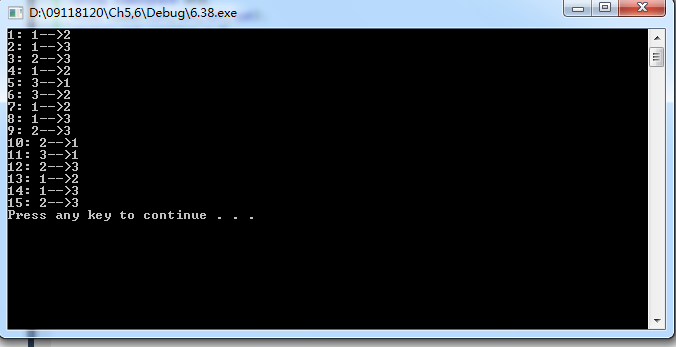
counter++;

cout << counter << ": " << start << "-->" << end << endl;

tower(num - 1, temp, start, end);

}

}



**6.37**

#include <iostream>

using namespace std;

int fibonacci(int);

int main()

{

int n;

while (cin) {

cout << "Type in a number: ";

cin >> n;

cout << "Fibonacci(" << n << ") = " << fibonacci(n) << endl;

}

}

int fibonacci(int n) {

if (n <= 0) {

//输入检查

cout << "Illegal Input!" << endl;

return 0;

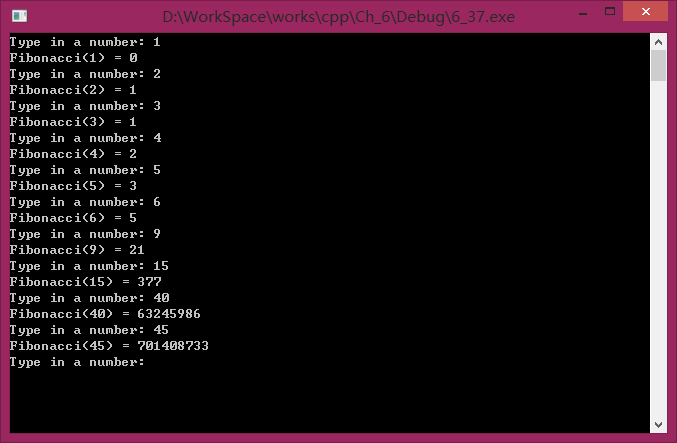
}

if (n <= 2)

//f(1) = 0, f(2) = 1

return n - 1;

int temp1 = 0, temp2 = 1, temp3;//temp1 = f(n-1), temp2 = f(n), temp3用于暂存

 for (int i = 0; i < n - 2; i++) {

temp3 = temp2;

temp2 += temp1;

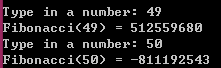
temp1 = temp3;

}

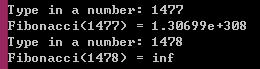
return temp2;

}

**b)** 最大能表示的数为512559680



**c)** 使用double后最大数为1.30699e+308



**6.41**

#include <iostream>

using namespace std;

int gcd(int, int);

int main() {

int n1, n2;

while (cin) {

cout << "Type 2 numbers(first one larger): ";

cin >> n1 >> n2;

cout << "The GCD of them is " << gcd(n1, n2) << endl;

}

}

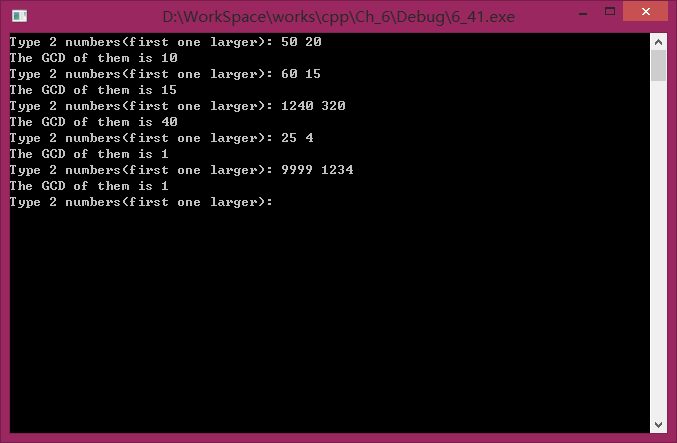
int gcd(int n1, int n2) {

if (n2 == 0)

return n1;

return gcd(n2, n1 % n2);

}



**6.46**

#include <iostream>

using namespace std;

int mystery(int, int);

int main() {

int x, y;

cout << "Enter 2 integers: ";

cin >> x >> y;

cout << "The result is " << mystery(x, y) << endl;

}

int mystery(int a, int b) {

if (b < 0)

return -mystery(a, -b);

if (b == 0)

return 0;

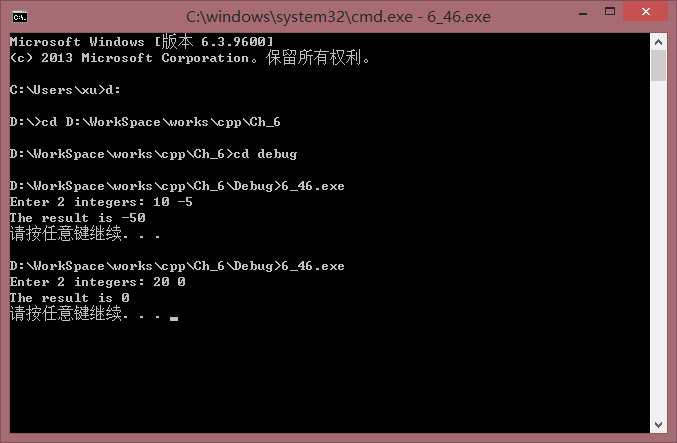
if (b == 1)

return a;

else

return a + mystery(a, b - 1);

}



**6.51**

#include <iostream>

using namespace std;

int tripleByValue(int);

void tripleByReference(int&);

int main() {

int count;

cout << "Type in a number: ";

cin >> count;

cout << "Triple by value: " << tripleByValue(count) << endl;

tripleByReference(count);

cout << "Triple by reference: " << count << endl;

system("pause");

}

int tripleByValue(int n) {

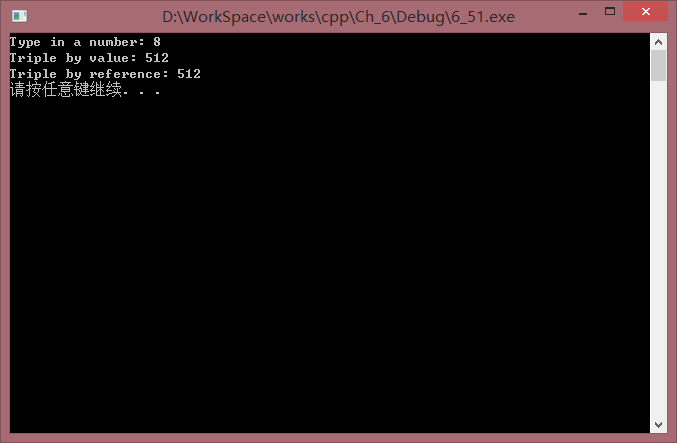
return n \* n \* n;

}

void tripleByReference(int& n) {

n = n \* n \* n;

}



**6.53**

//minimum.h

template<typename T>

T minimum(T value1, T value2, T value3) {

T temp = value1;

if (temp > value2)

temp = value2;

if (temp > value3)

temp = value3;

return temp;

}

#include <iostream>

#include "minimum.h"

using namespace std;

int main() {

int a, b, c;

cout << "Type in 3 integers: ";

cin >> a >> b >> c;

cout << "The minimum is " << minimum(a, b, c) << endl;

char x, y, z;

cout << "Type in 3 characters: ";

cin >> x >> y >> z;

cout << "The minimum is " << minimum(x, y, z) << endl;

double d1, d2, d3;

cout << "Type in 3 floating-point numbers: ";

cin >> d1 >> d2 >> d3;

cout << "The minimum is " << minimum(d1, d2, d3) << endl;

system("pause");

}

