

C语言程序设计

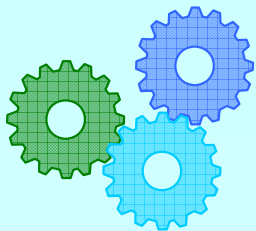
李 祥

湖北经济学院信息管理学院

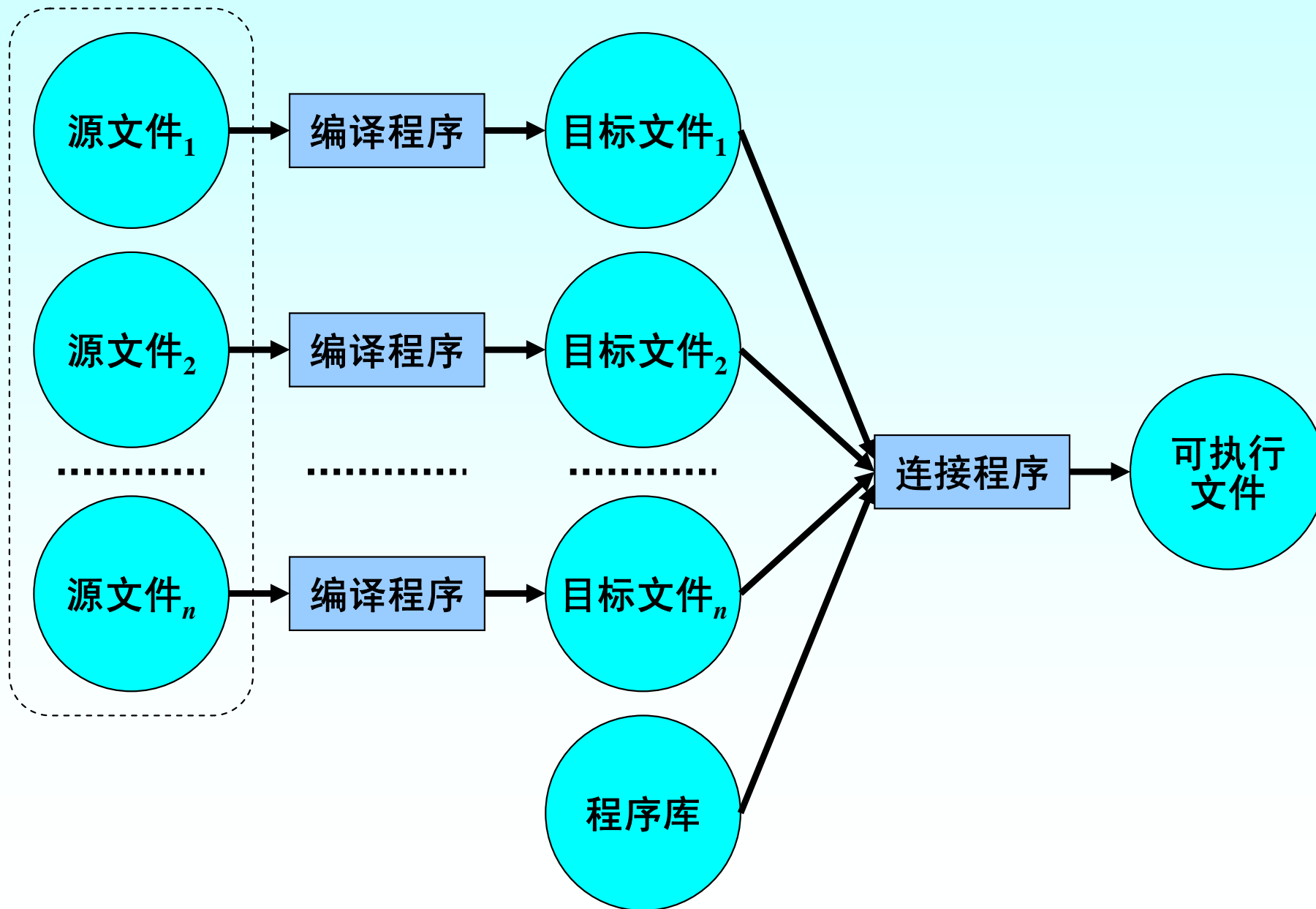
2014.12

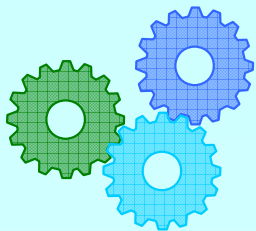


01100100101101001

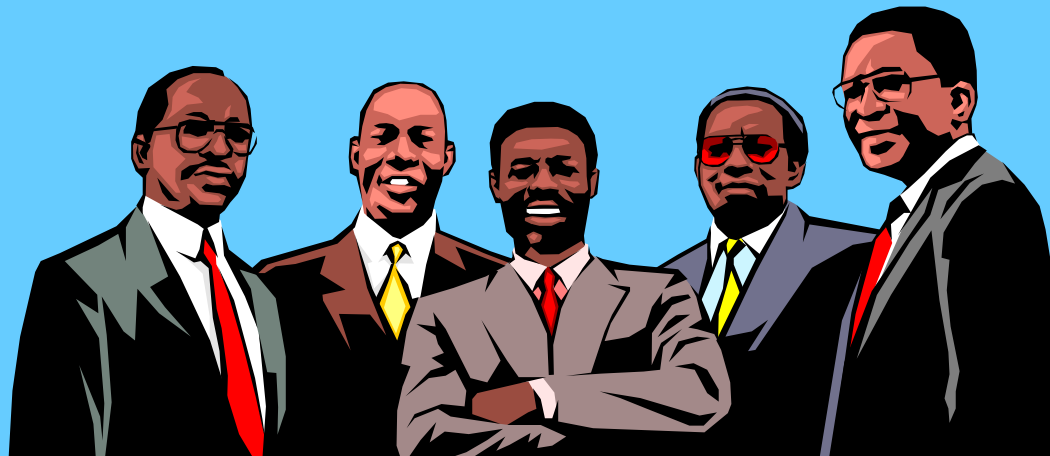


编译和连接

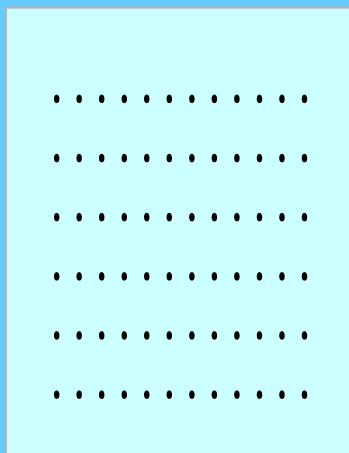




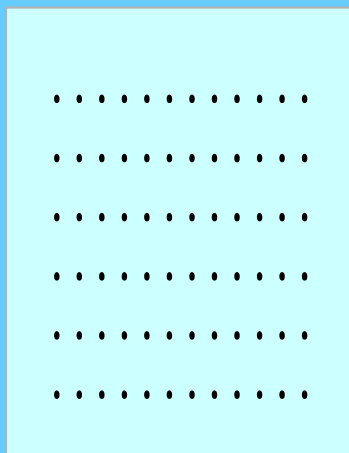
Dev-C++工程项目的组织形式



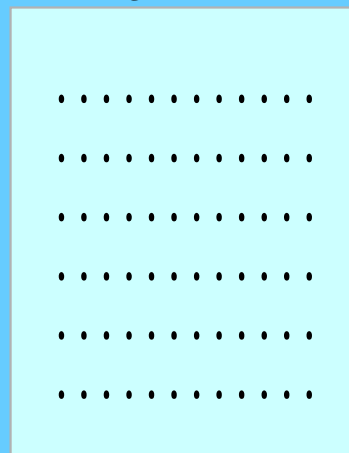
文件₁



文件₂

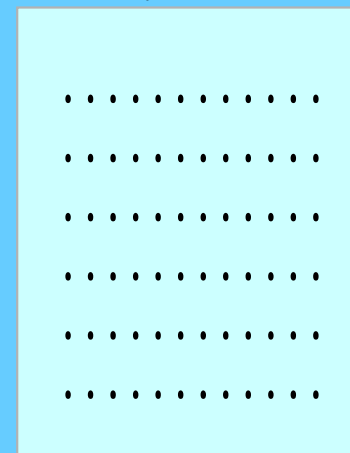


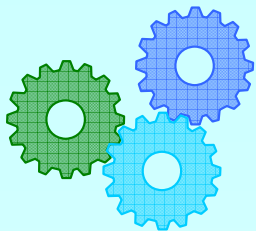
文件₃



...

文件_n

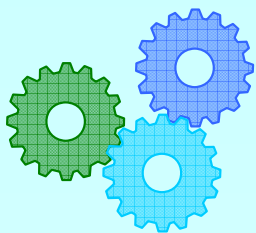




Dev-C++ 基本操作步骤

1. 创建工程项目文件夹
2. 在该文件夹中创建工程项目文件(.dev)
3. 系统自动创建源程序文件main.c及主函数, 修改并保存该文件
4. 打开项目视图, 在项目中新建源程序文件(.c), 输入程序并保存该文件
5. 对各源程序文件分别进行编译(Ctrl+F9), 得到目标文件(.o)
6. 将目标文件和系统库进行连接(F9), 得到可执行文件(.exe)
7. 运行可执行文件, 输入数据, 检查输出结果





Dev-C++工程项目组织形式

工程项目(.dev)

源文件₁(.c)

全局说明

类型 函数(参数表)

```
{  
  ...  
}
```

类型 函数(参数表)

```
{  
  ...  
}
```

...

类型 函数(参数表)

```
{  
  ...  
}
```

源文件₂(.c)

全局说明

类型 函数(参数表)

```
{  
  ...  
}
```

类型 函数(参数表)

```
{  
  ...  
}
```

...

类型 函数(参数表)

```
{  
  ...  
}
```

源文件_n(.c)

全局说明

类型 函数(参数表)

```
{  
  ...  
}
```

类型 函数(参数表)

```
{  
  ...  
}
```

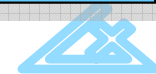
...

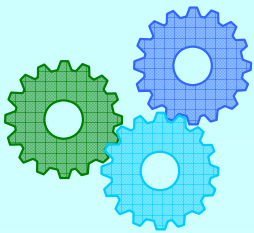
类型 函数(参数表)

```
{  
  ...  
}
```

通常一个工程项目由许多源程序文件组成。

每个源程序文件由许多函数组成。





Dev-C++工程项目组织形式

工程项目: Project.dev

源文件: main.c

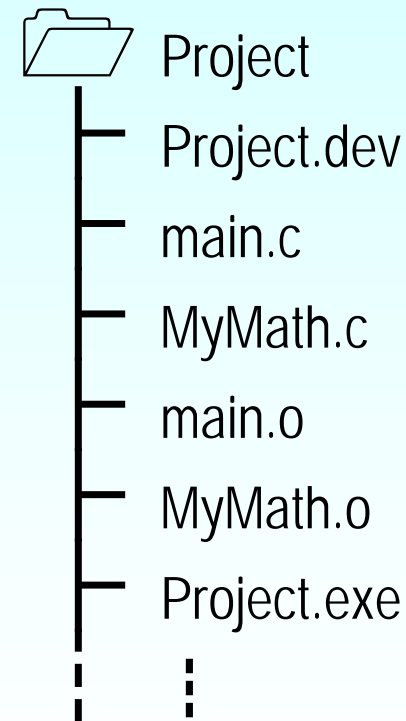
```
#include <stdio.h>

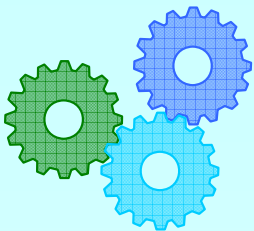
int Max(int x, int y);

int main()
{
    int a, b, c;
    printf("a = ? ");
    scanf("%d", &a);
    printf("b = ? ");
    scanf("%d", &b);
    c = Max(a, b);
    printf("c = %d\n", c);
    return 0;
}
```

源文件: MyMath.c

```
int Max(int x, int y)
{
    int z;
    if (x >= y)
    {
        z = x;
    }
    else
    {
        z = y;
    }
    return z;
}
```

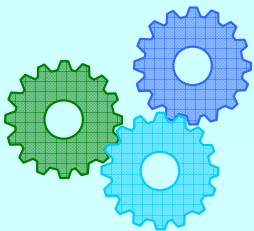




Dev-C++ 常用快捷键

- 文件操作
 - 新建文件: Ctrl + N
 - 打开文件: Ctrl + O
 - 保存文件: Ctrl + S
 - 关闭文件: Ctrl + W
- 文件卡操作
 - 下一卡片: Ctrl + Tab
 - 上一卡片: Shift + Ctrl + Tab
 - 切换头/源文件卡片: Ctrl + Q
- 编辑操作
 - 增大缩进: Tab
 - 减小缩进: Shift + Tab
 - 添加注释: Ctrl + .
 - 取消注释: Ctrl + ,
 - 设置/清除书签: Ctrl + 1 ~ 9
 - 跳到书签: Alt + Ctrl + 1 ~ 9
- 查找和替换
 - 查找: Ctrl + F
 - 替换: Ctrl + R
- 编译、连接和运行
 - 编译: Ctrl + F9
 - 编译和连接: F9
 - 运行: F10
 - 编译、连接和运行: F11
 - 全部重新编译并连接: F12
 - 强制结束运行: Ctrl + Break
- 调试程序
 - 调试运行: F5
 - 设置/清除断点: F4
 - 下一步: F7
 - 深入函数: F8
 - 结束调试: F6





最大公约数

辗转相除法(欧几里德算法)

m	\bmod	n	\rightarrow	r
36		21		15
21		15		6
15		6		3
6		3		0

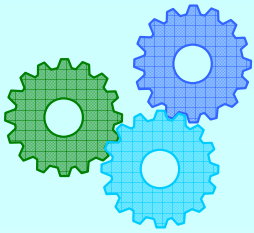
m	\bmod	n	\rightarrow	r
14		34		14
34		14		6
14		6		2
6		2		0

当 m 、 n 至少有一个为零时, 显然, $\text{Gcd}(36, 0)$ 的值为36; $\text{Gcd}(0, 21)$ 的值为21; 为方便处理, 姑且规定 $\text{Gcd}(0, 0)$ 的值为1。

当 m 、 n 均不为零时, 设 $r = m \bmod n$, 若 $r = 0$, 则 $\text{Gcd}(m, n)$ 的值为 n ; 若 $r \neq 0$, 则 $\text{Gcd}(m, n)$ 的值与 $\text{Gcd}(n, r)$ 的值相同。

```
int Gcd(int m, int n)
{
    int r, d;
    if (0 == m && 0 == n)
    {
        d = 1;
    }
    else if (0 == m)
    {
        d = n;
    }
    else if (n == 0)
    {
        d = m;
    }
    else
    {
        r = m % n;
        if (0 == r)
        {
            d = n;
        }
        else
        {
            d = Gcd(n, r);
        }
    }
    return d;
}
```





最小公倍数

$$36 = 2 \times 2 \times 3 \times 3$$

$$21 = 3 \times 7$$

$$\text{Lcm}(36, 21) = 36 \times 21 \div 3 = 252$$

$$14 = 2 \times 7$$

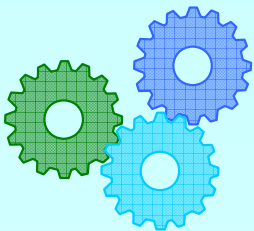
$$34 = 2 \times 17$$

$$\text{Lcm}(14, 34) = 14 \times 34 \div 2 = 238$$

$$\text{Lcm}(m, n) = m \times n \div \text{Gcd}(m, n)$$

```
int Lcm(int m, int n)
{
    return m * n / Gcd(m, n);
}
```





多文件程序

MyMath.c

```
int Gcd(int m, int n);
int Lcm(int m, int n);

int Gcd(int m, int n)
{
    int r, d;
    if (0 == m && 0 == n)
    {
        d = 1;
    }
    else if (0 == m)
    {
        d = n;
    }
    else if (n == 0)
    {
        d = m;
    }
    else
    {
        r = m % n;
        if (0 == r)
        {
            d = n;
        }
        else
        {
            d = Gcd(n, r);
        }
    }
    return d;
}

int Lcm(int m, int n)
{
    return m * n / Gcd(m, n);
}
```

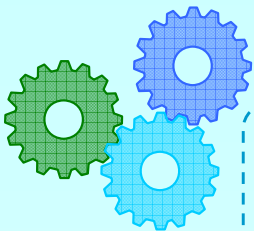
main.c

```
#include <stdio.h>

int Gcd(int m, int n);
int Lcm(int m, int n);

int main()
{
    int a, b, c, d;
    printf("两个自然数: ");
    scanf("%d%d", &a, &b);
    c = Gcd(a, b);
    d = Lcm(a, b);
    printf("最大公约数: %d\n", c);
    printf("最小公倍数: %d\n", d);
    return 0;
}
```





利用头文件

MyMath.h

```
int Gcd(int m, int n);  
int Lcm(int m, int n);
```

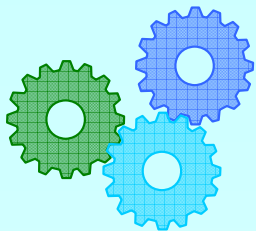
MyMath.c

```
#include "MyMath.h"  
  
int Gcd(int m, int n)  
{  
    int r, d;  
    if (0 == m && 0 == n)  
    {  
        d = 1;  
    }  
    else if (0 == m)  
    {  
        d = n;  
    }  
    else if (n == 0)  
    {  
        d = m;  
    }  
    else  
    {  
        r = m % n;  
        if (0 == r)  
        {  
            d = n;  
        }  
        else  
        {  
            d = Gcd(n, r);  
        }  
    }  
    return d;  
}  
  
int Lcm(int m, int n)  
{  
    return m * n / Gcd(m, n);  
}
```

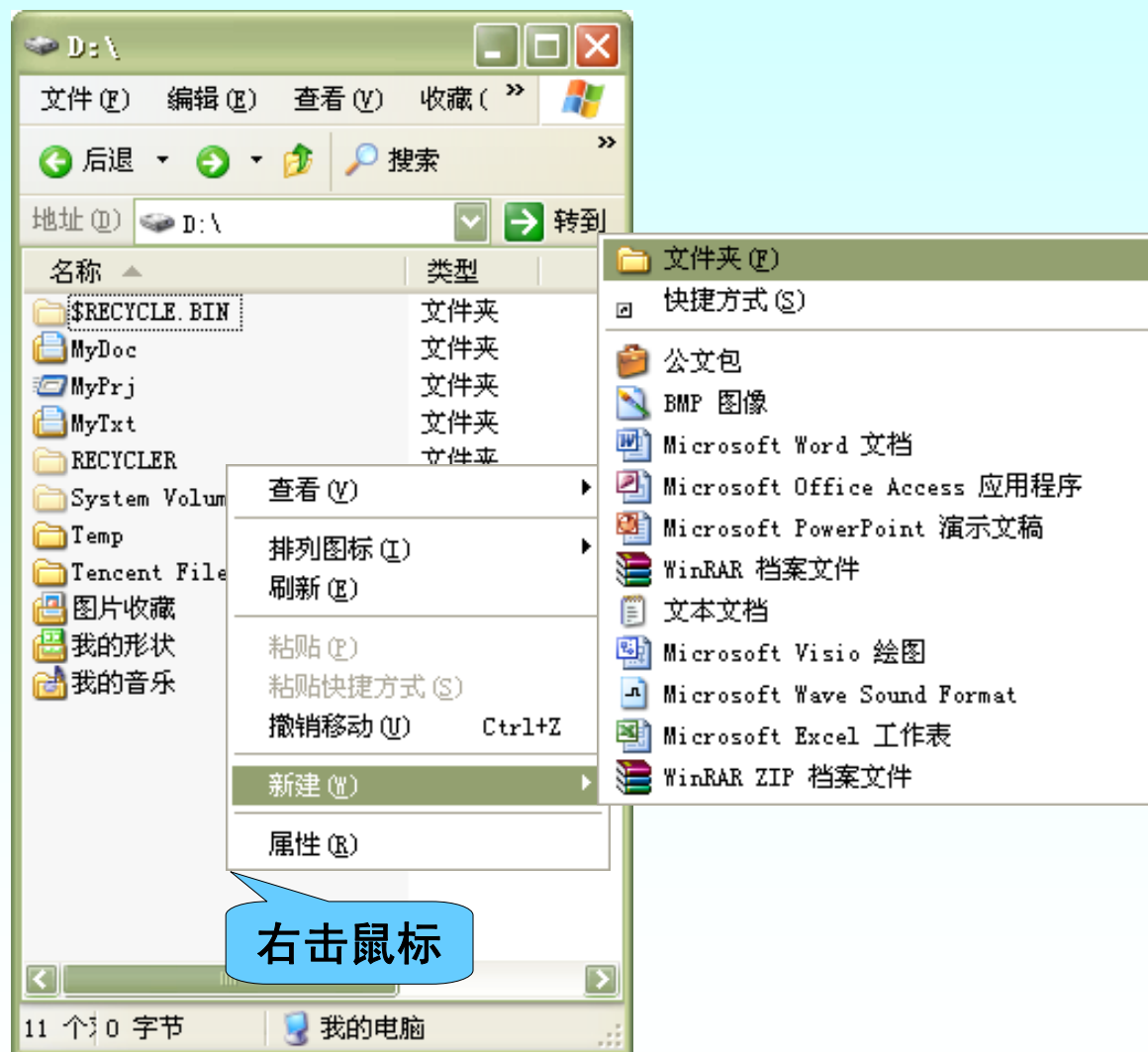
main.c

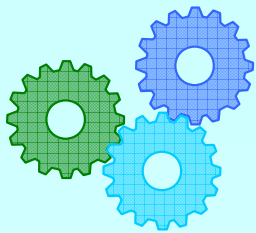
```
#include <stdio.h>  
#include "MyMath.h"  
  
int main()  
{  
    int a, b, c, d;  
    printf("两个自然数: ");  
    scanf("%d%d", &a, &b);  
    c = Gcd(a, b);  
    d = Lcm(a, b);  
    printf("最大公约数: %d\n", c);  
    printf("最小公倍数: %d\n", d);  
    return 0;  
}
```





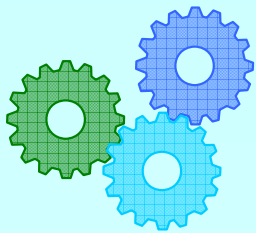
创建项目文件夹





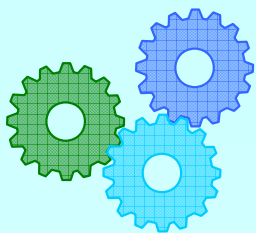
创建项目文件夹



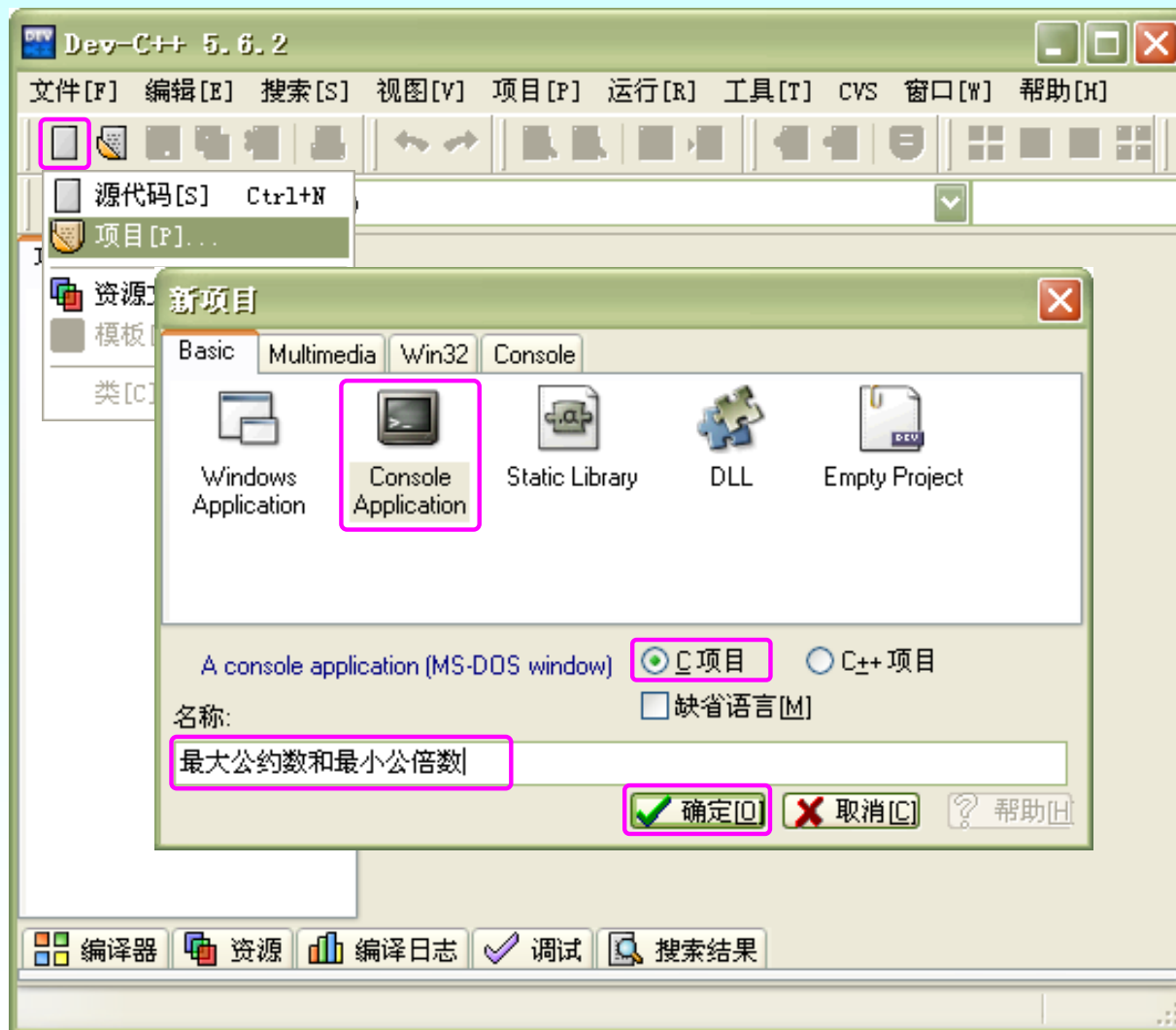


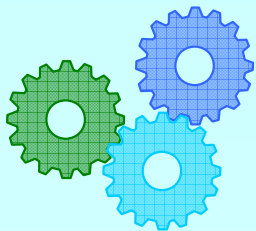
创建项目文件夹





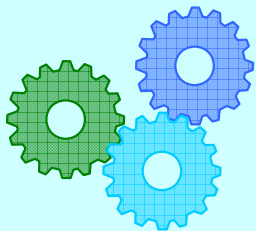
创建工程项目



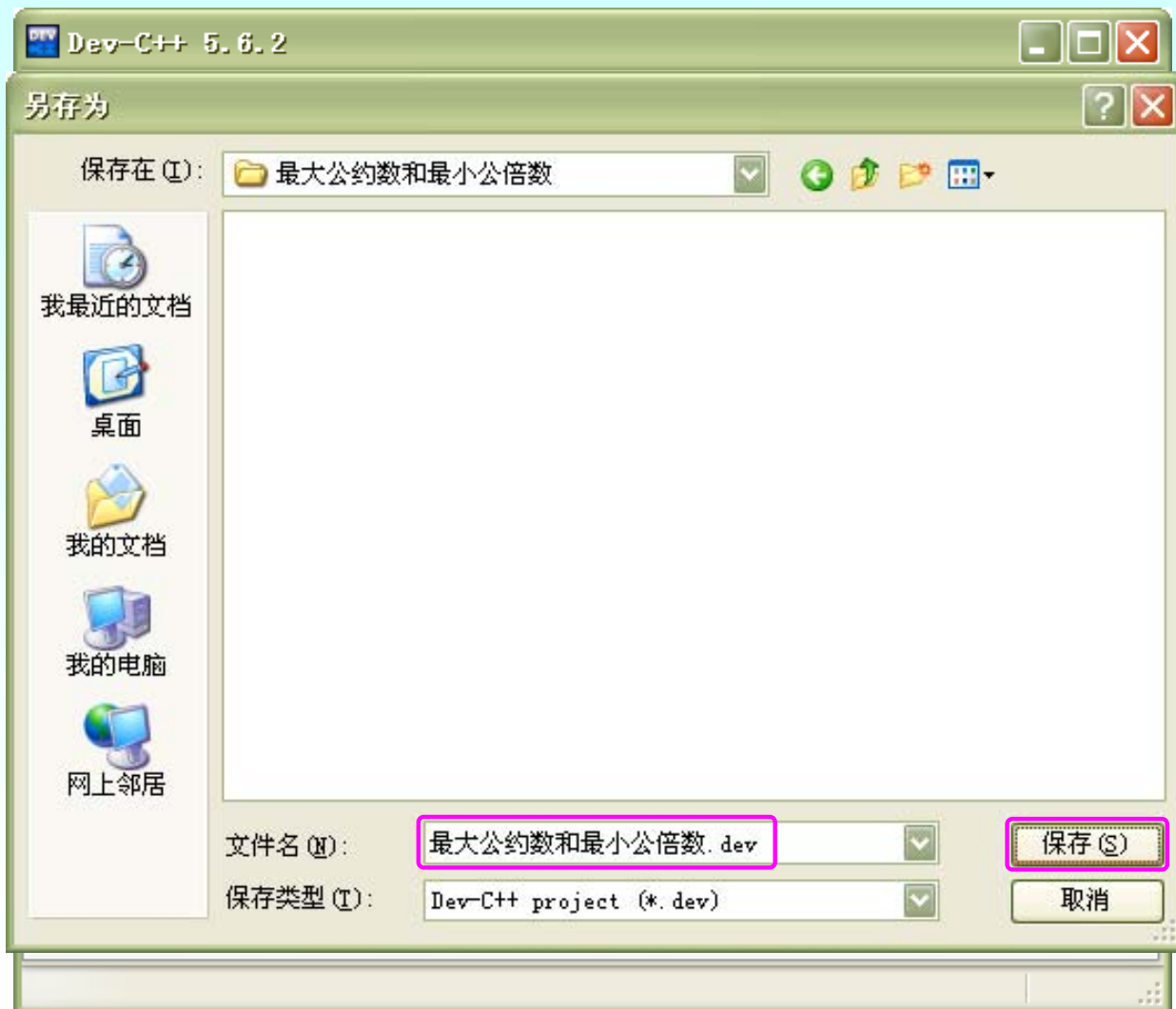


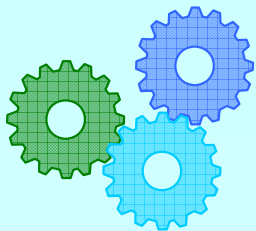
创建工程项目



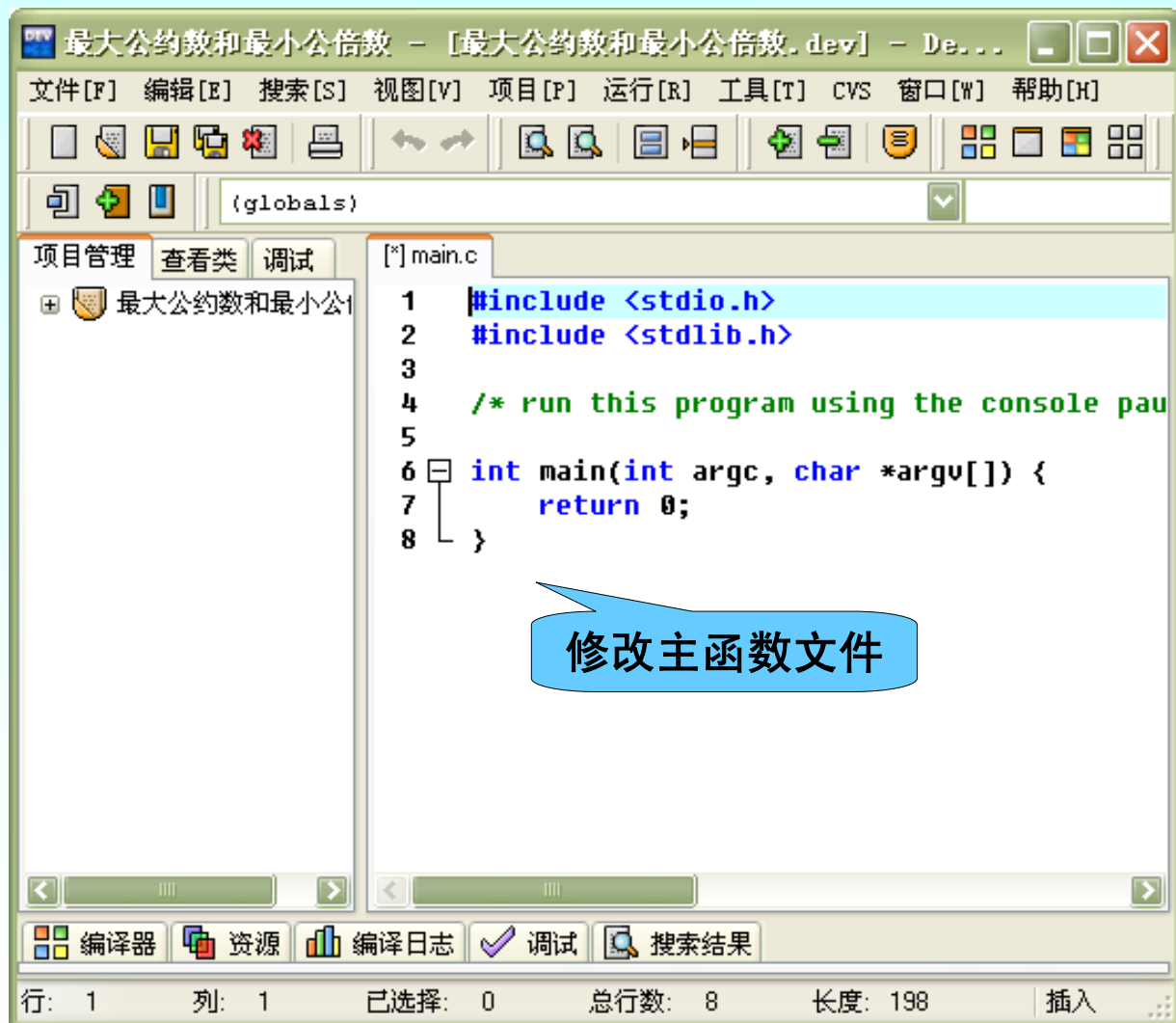
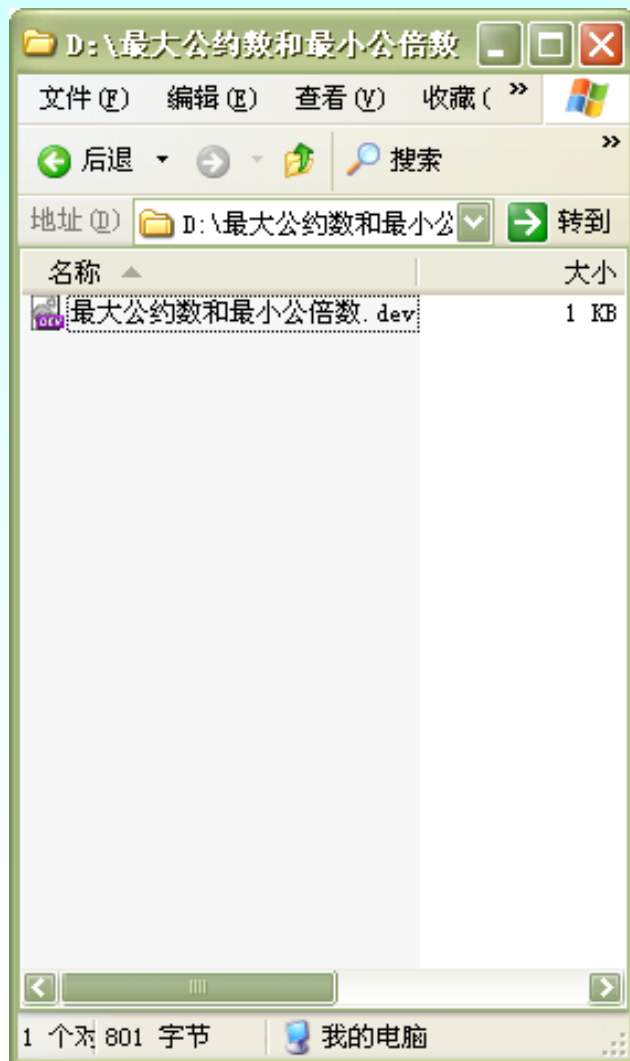


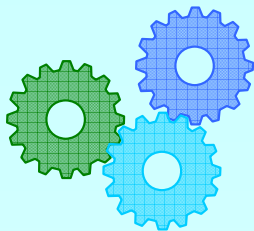
创建工程项目



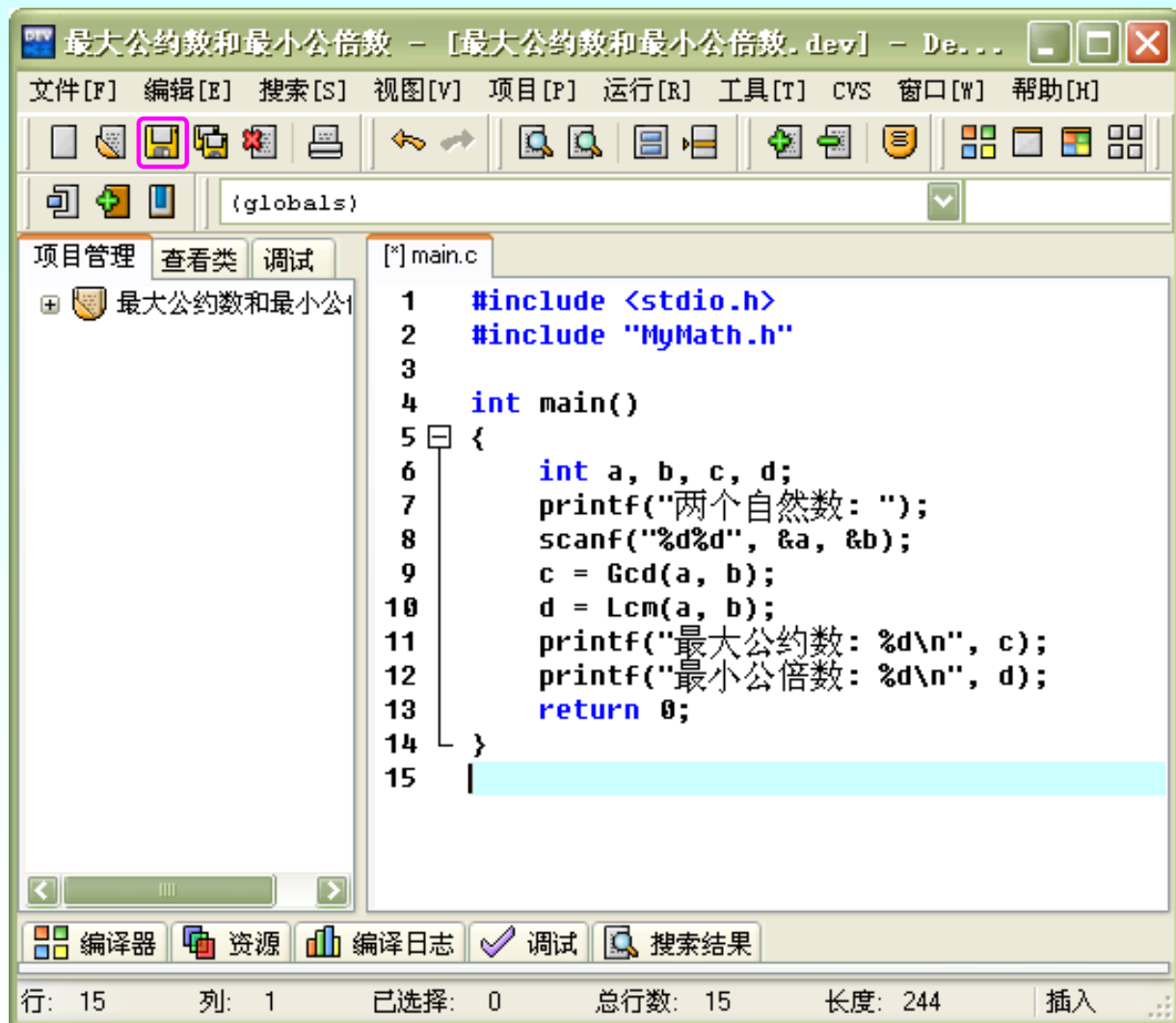
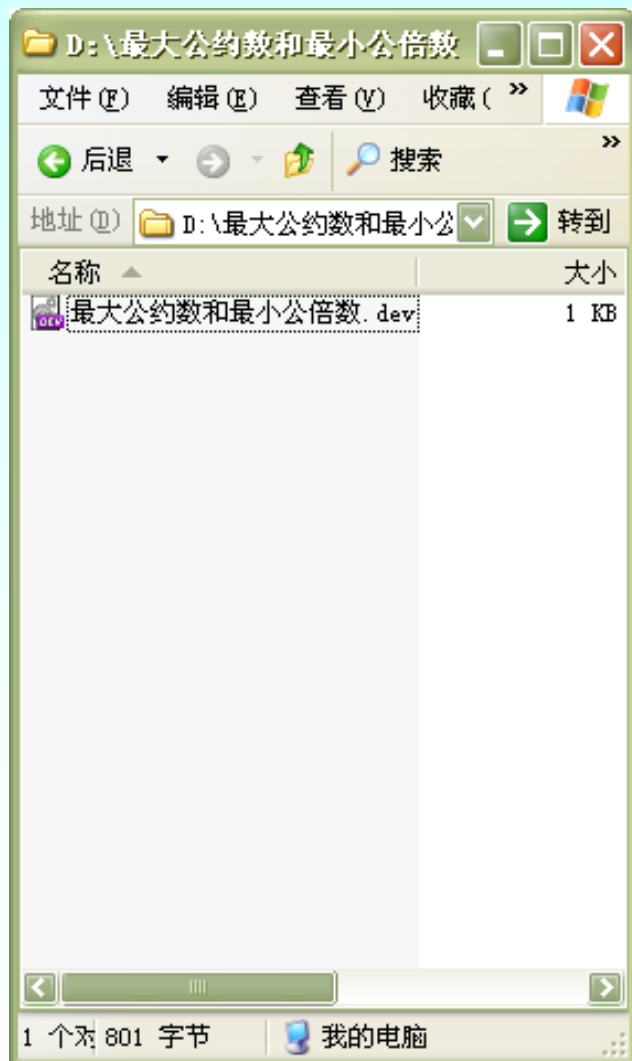


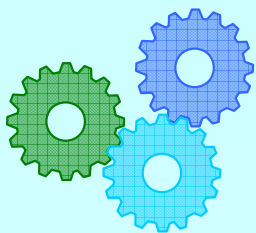
修改主函数文件



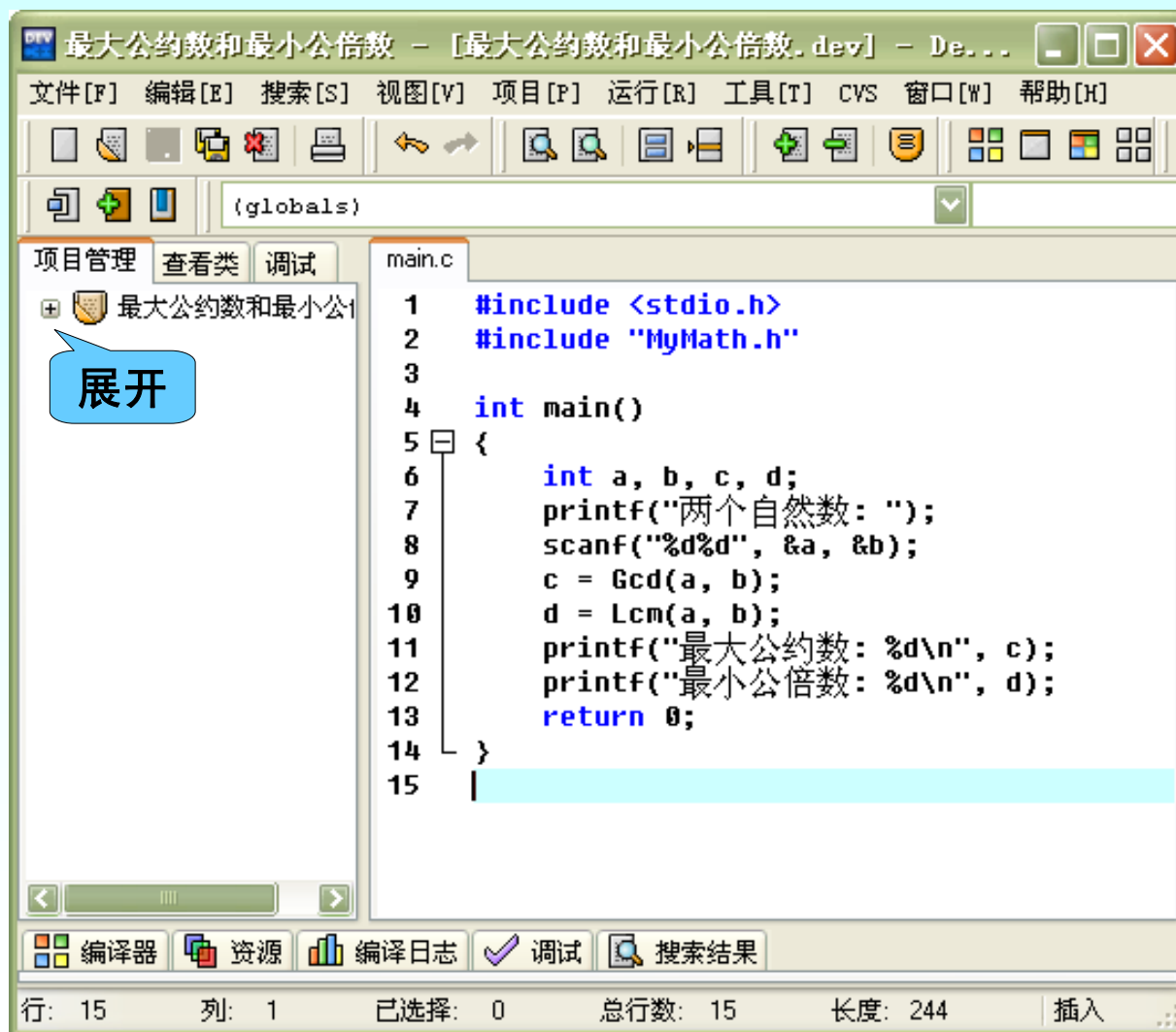
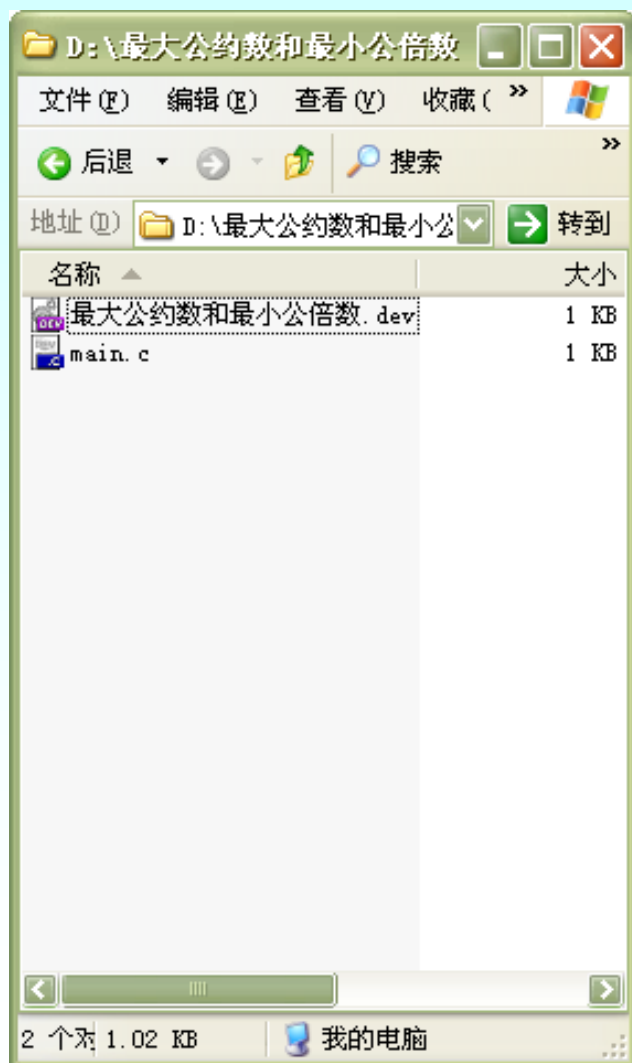


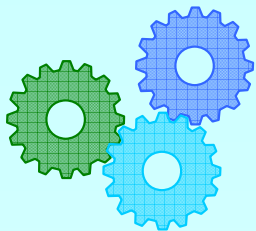
保存主函数文件



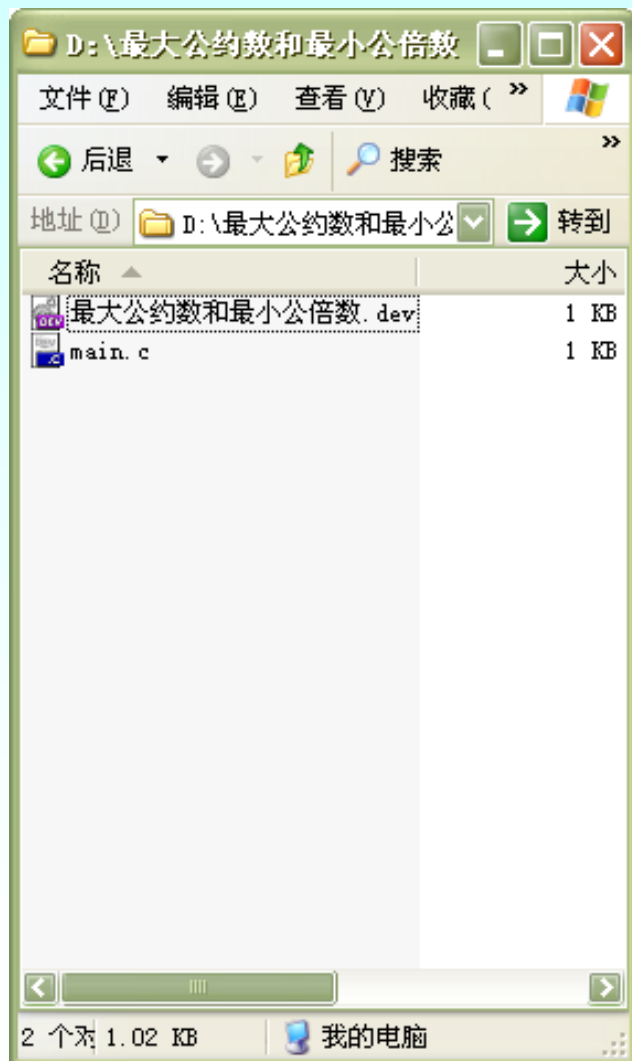


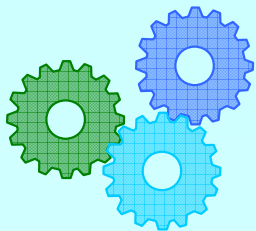
创建“我的数学”头文件



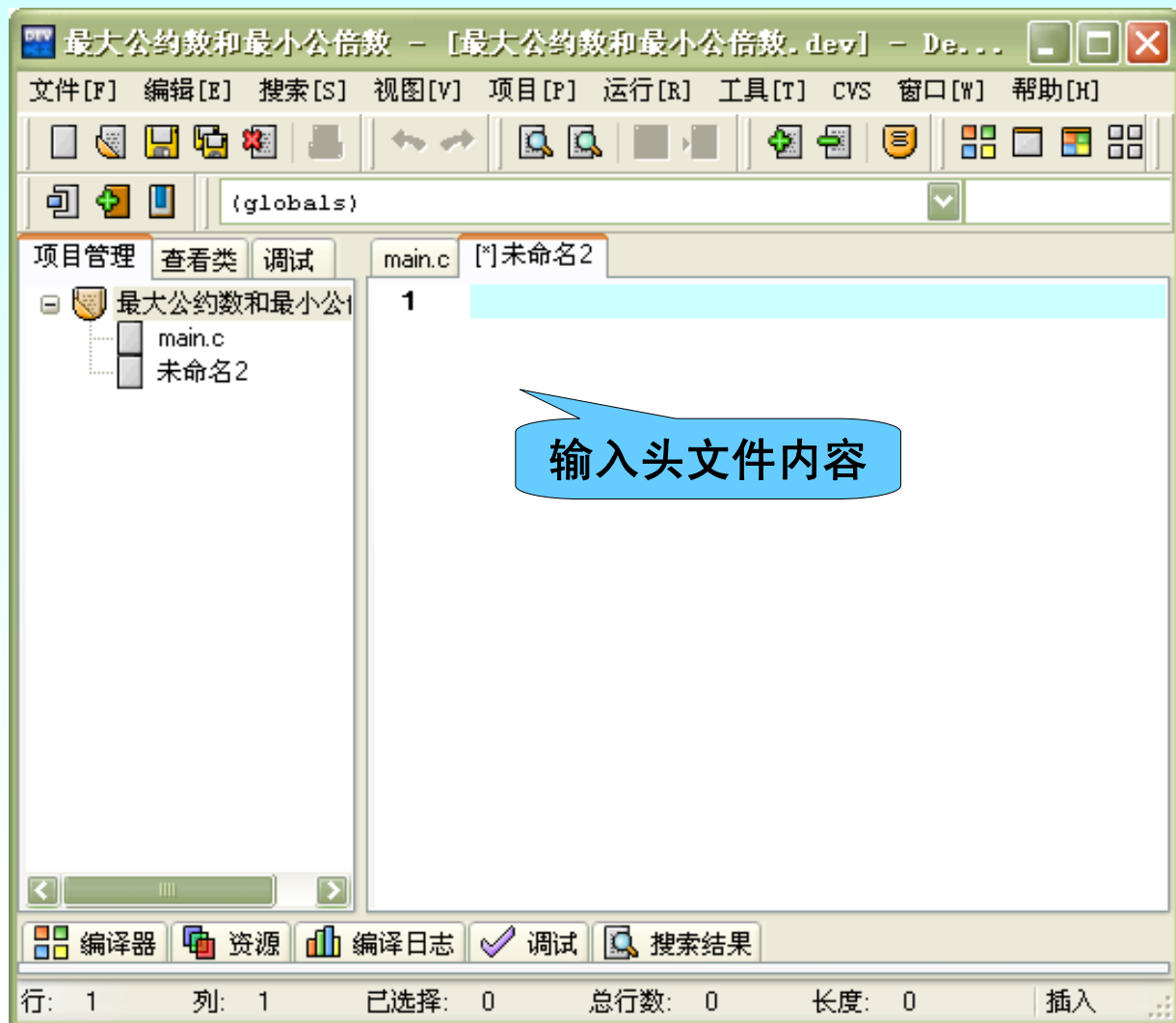
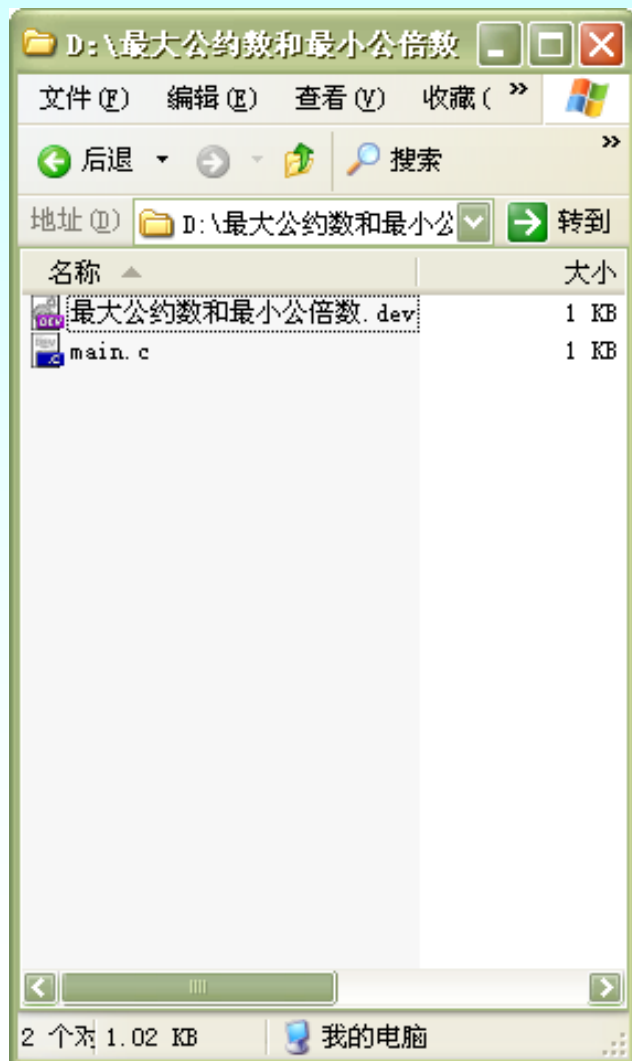


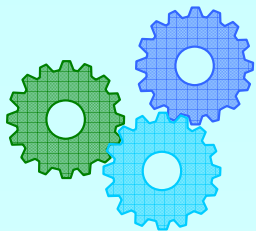
创建“我的数学”头文件



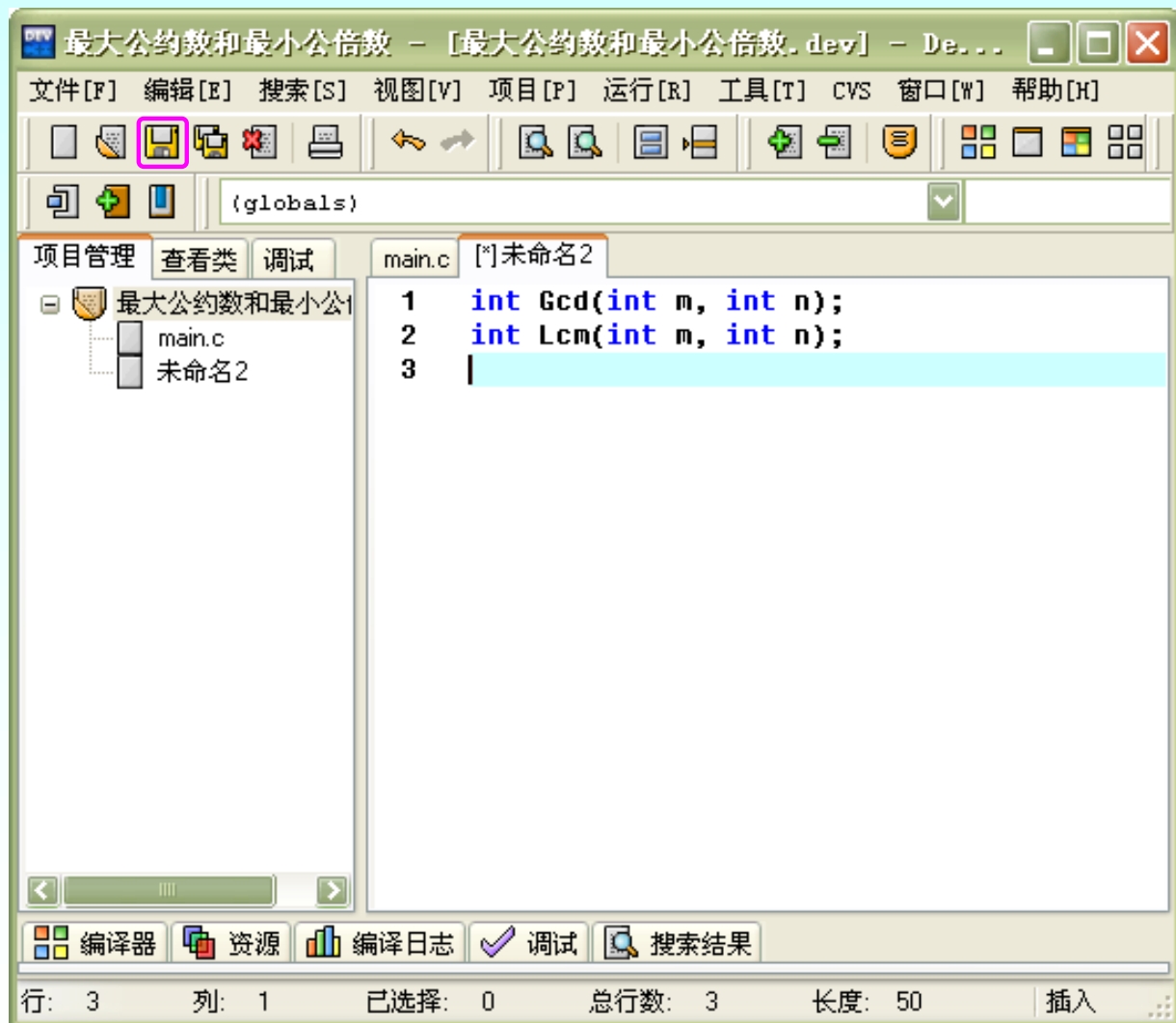
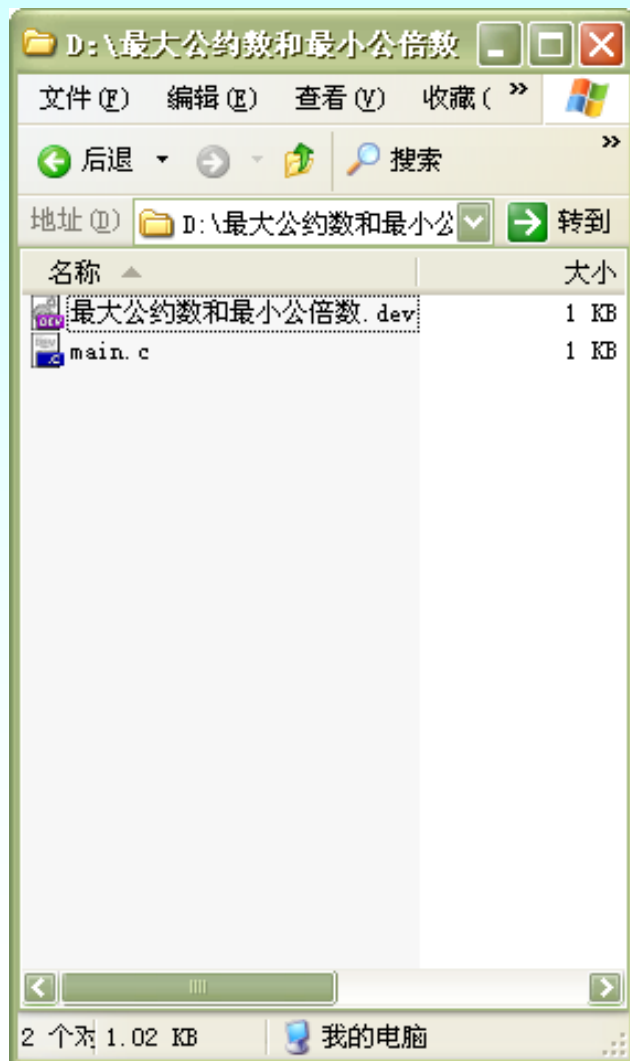


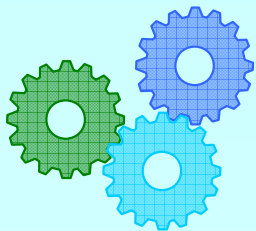
创建“我的数学”头文件



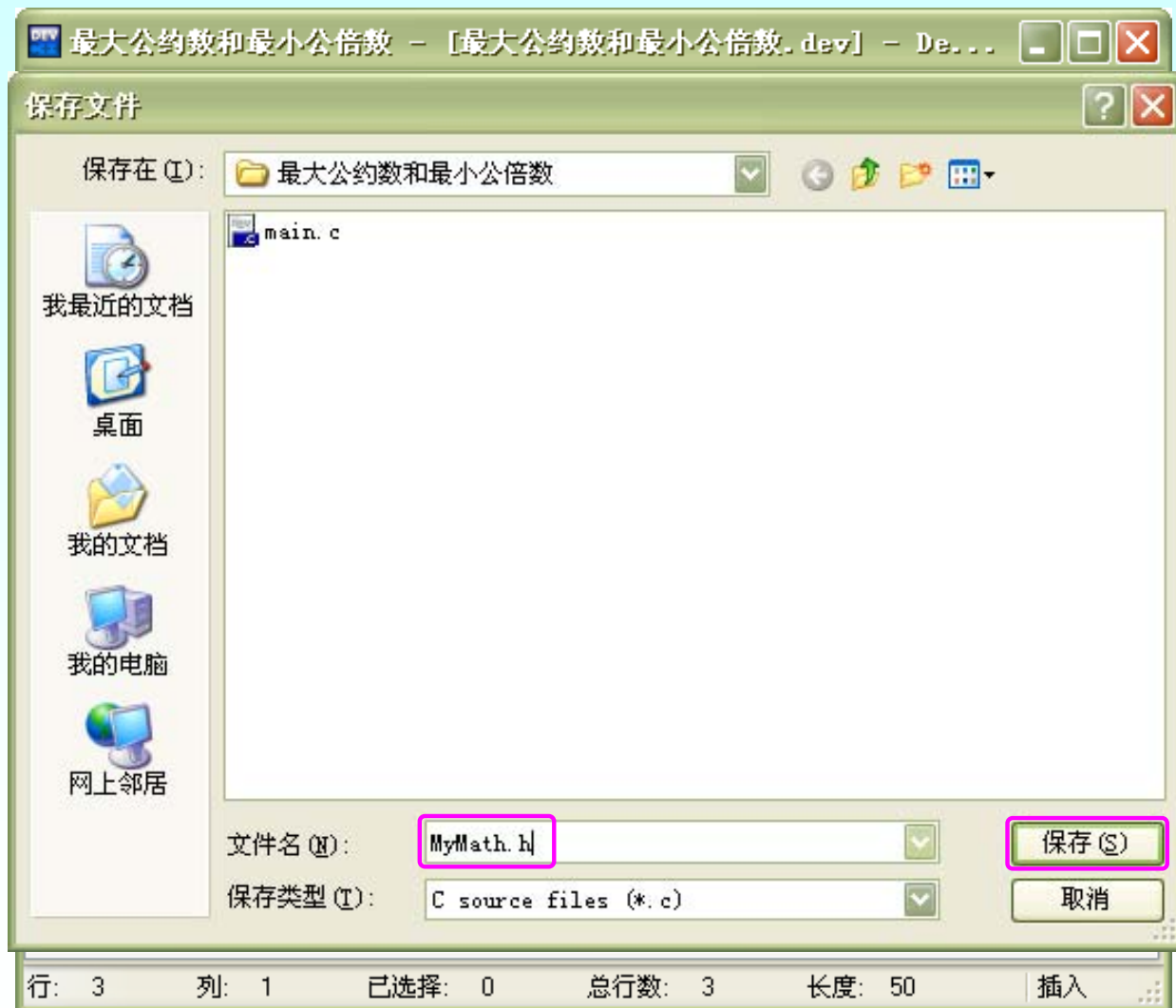
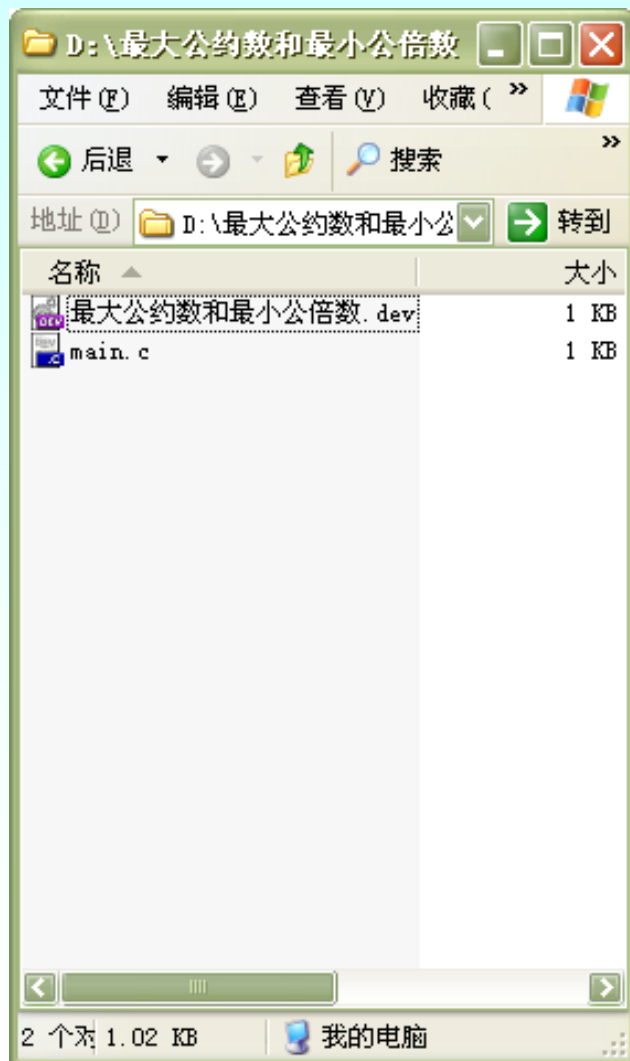


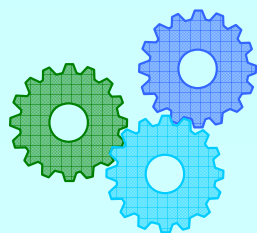
保存“我的数学”头文件



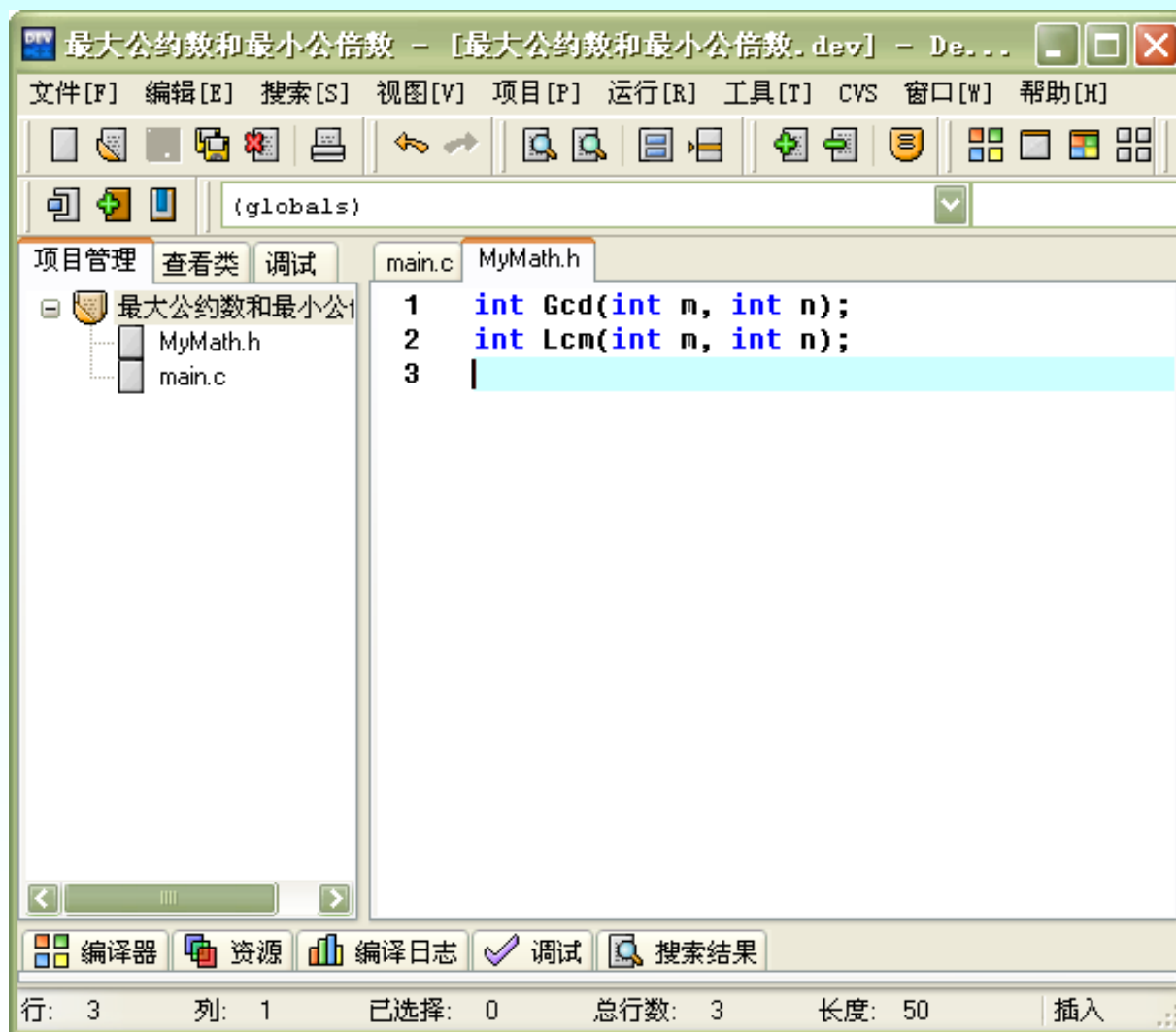
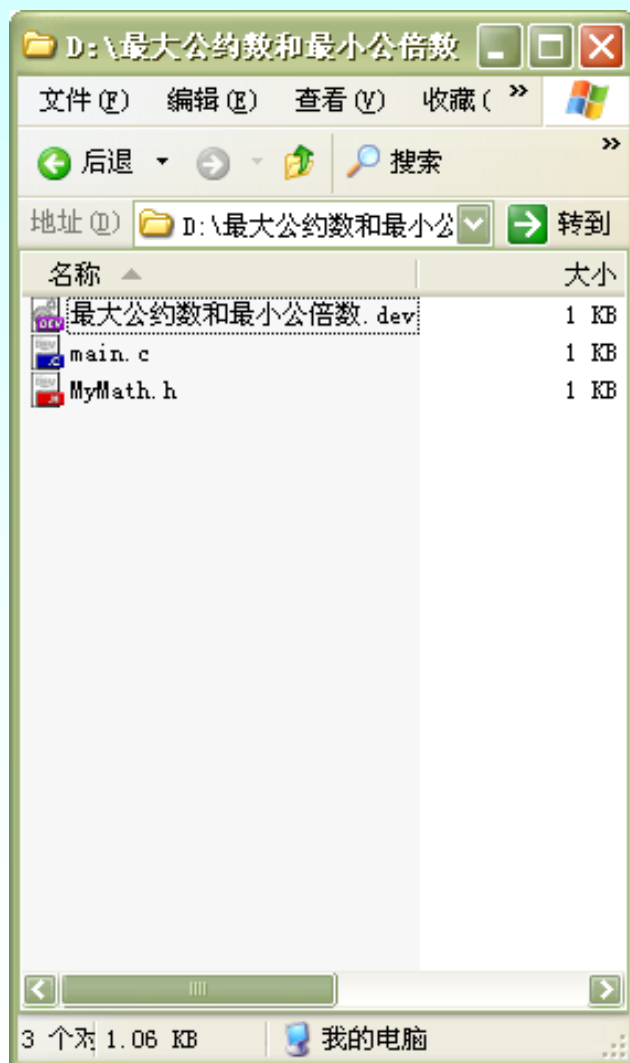


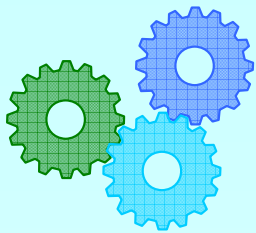
保存“我的数学”头文件



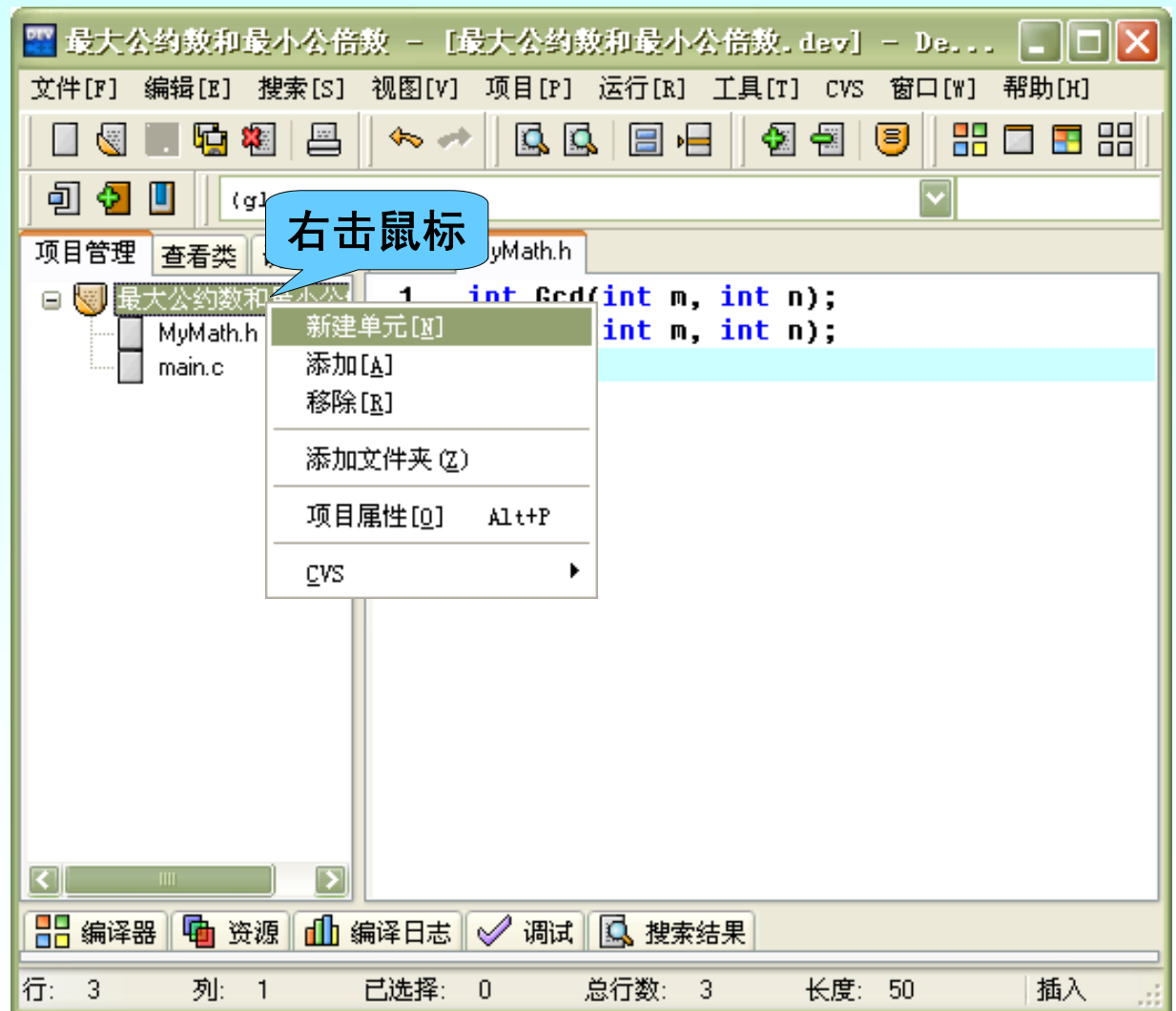
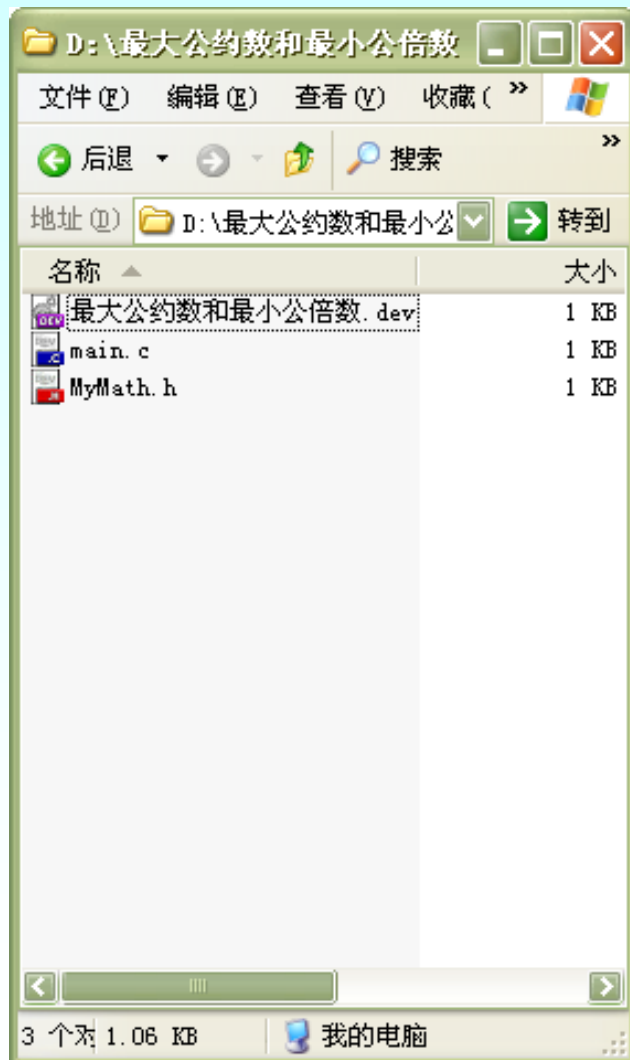


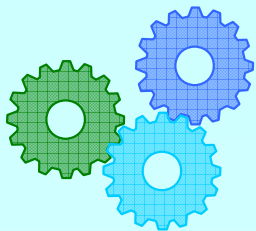
保存“我的数学”头文件



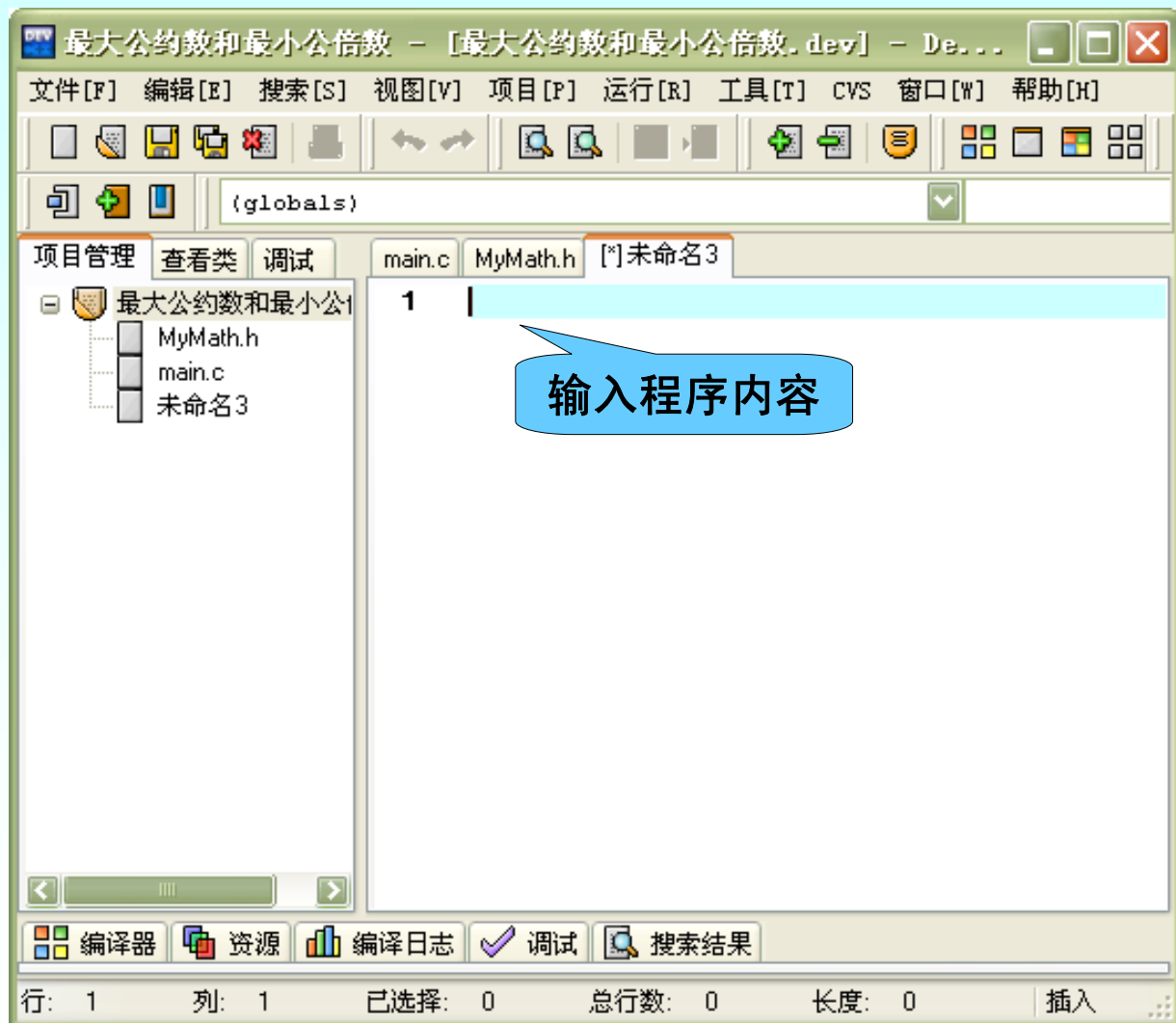
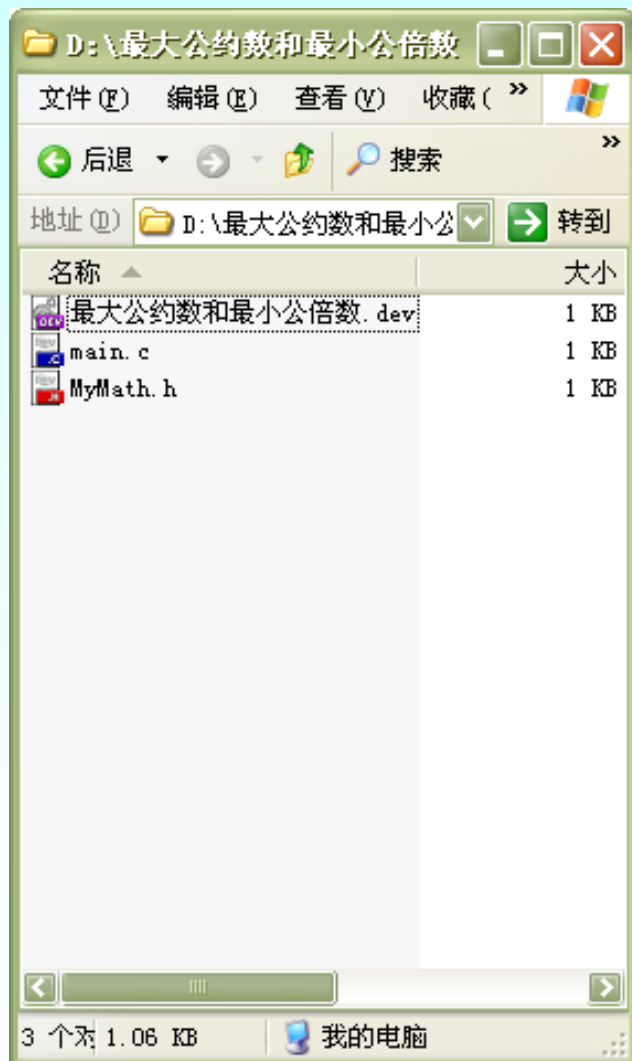


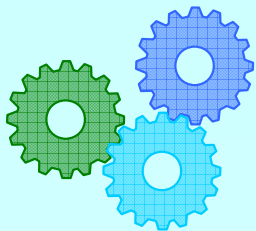
创建“我的数学”程序文件



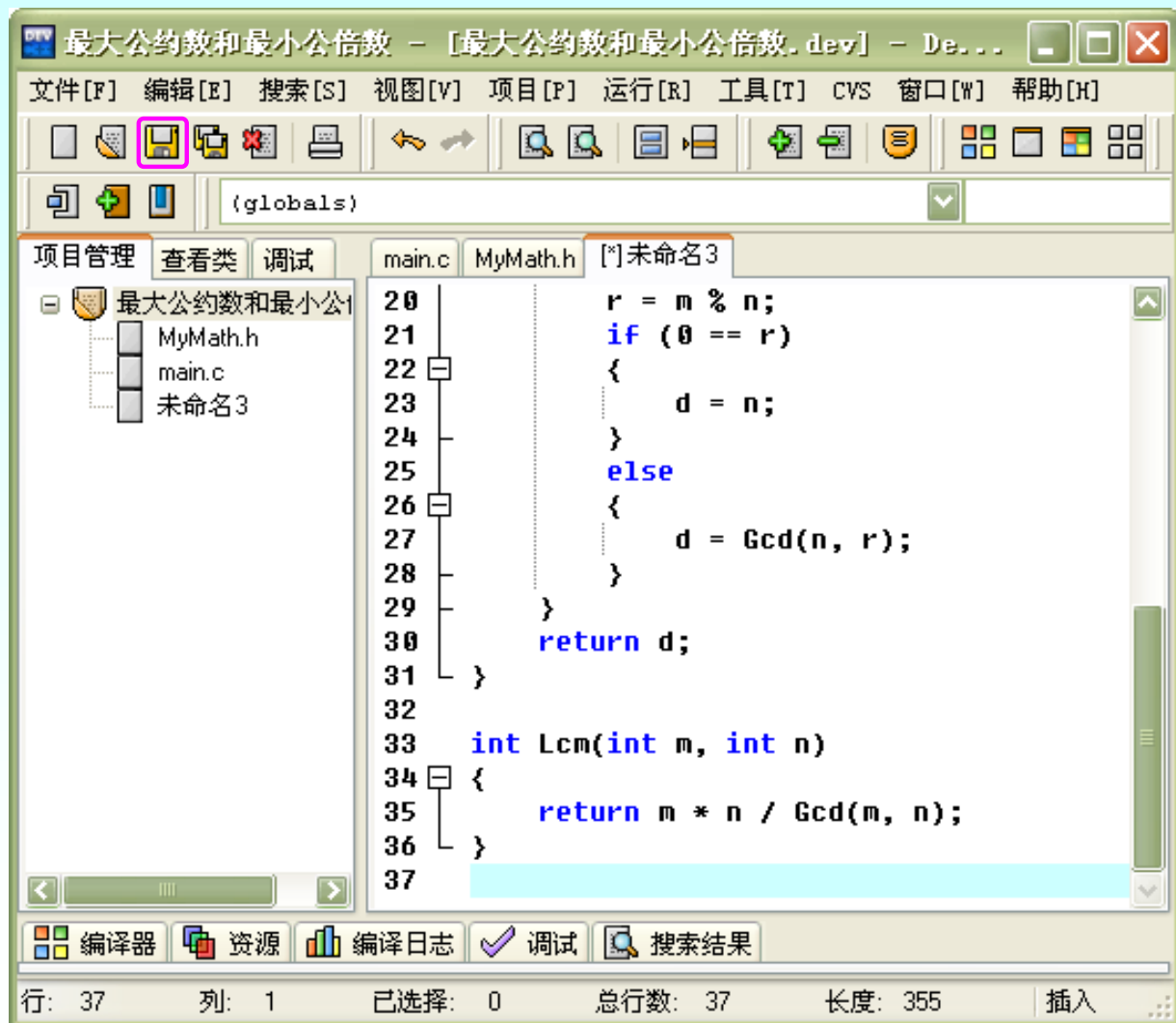
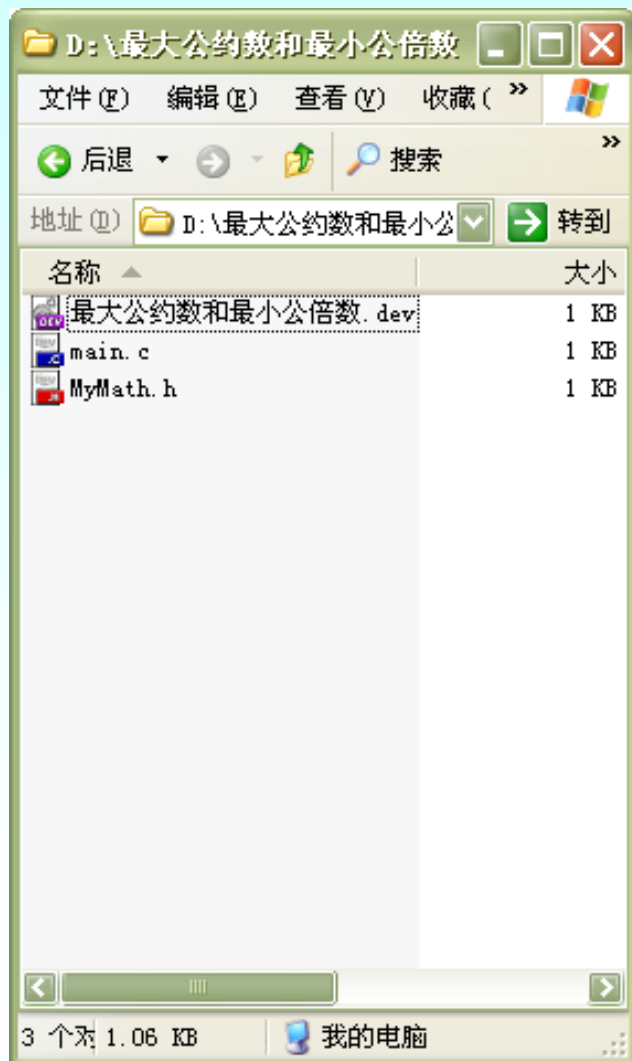


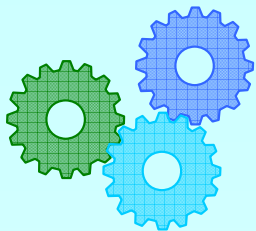
创建“我的数学”程序文件



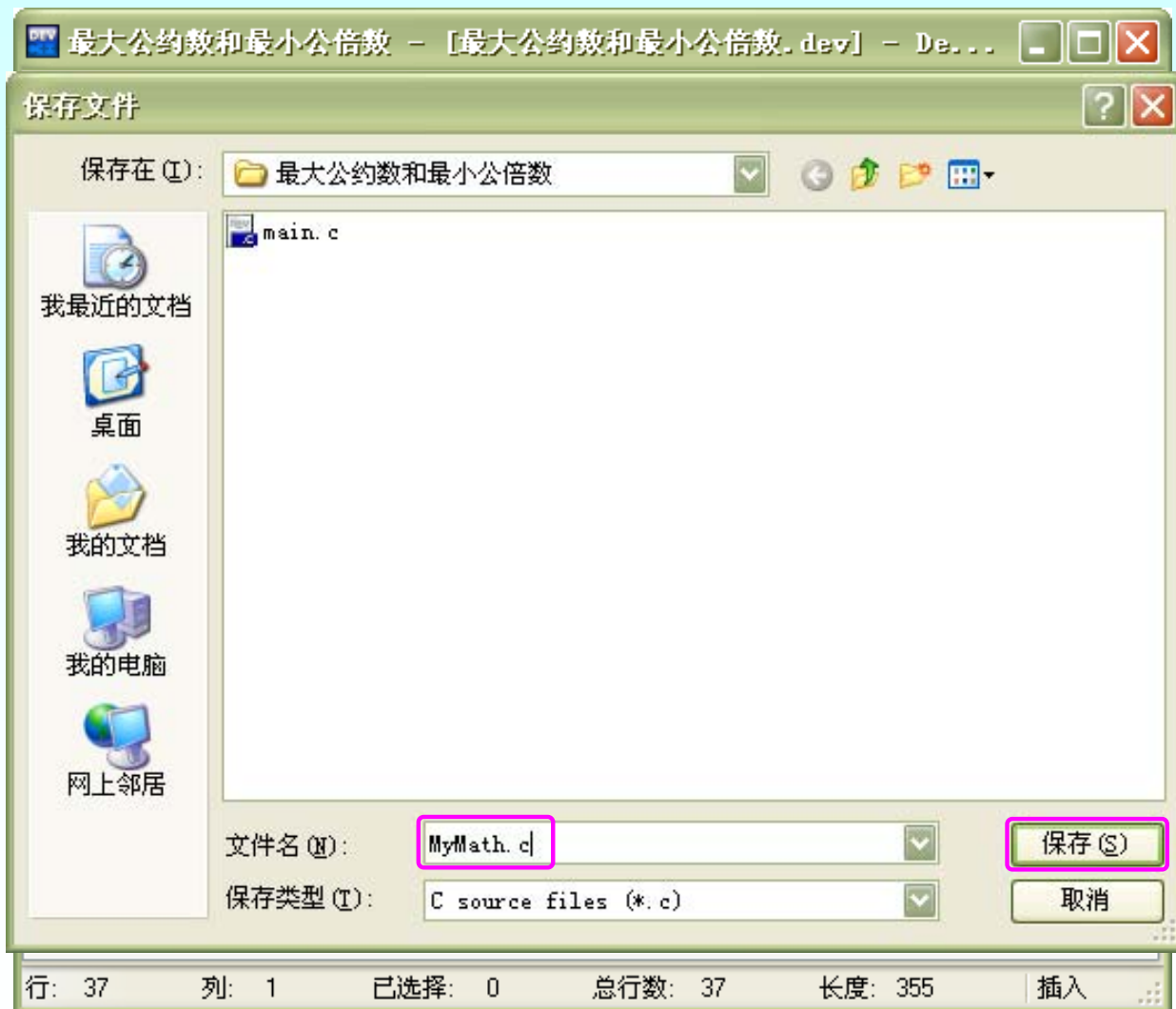
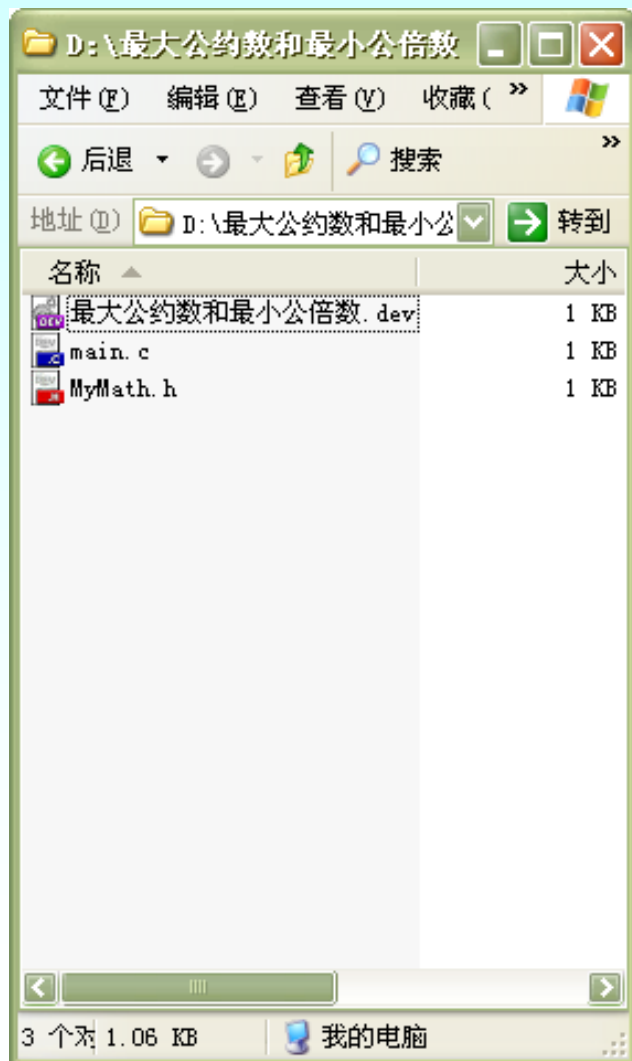


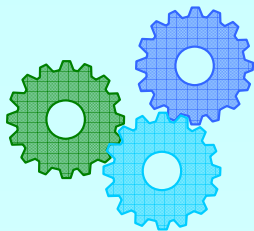
保存“我的数学”程序文件



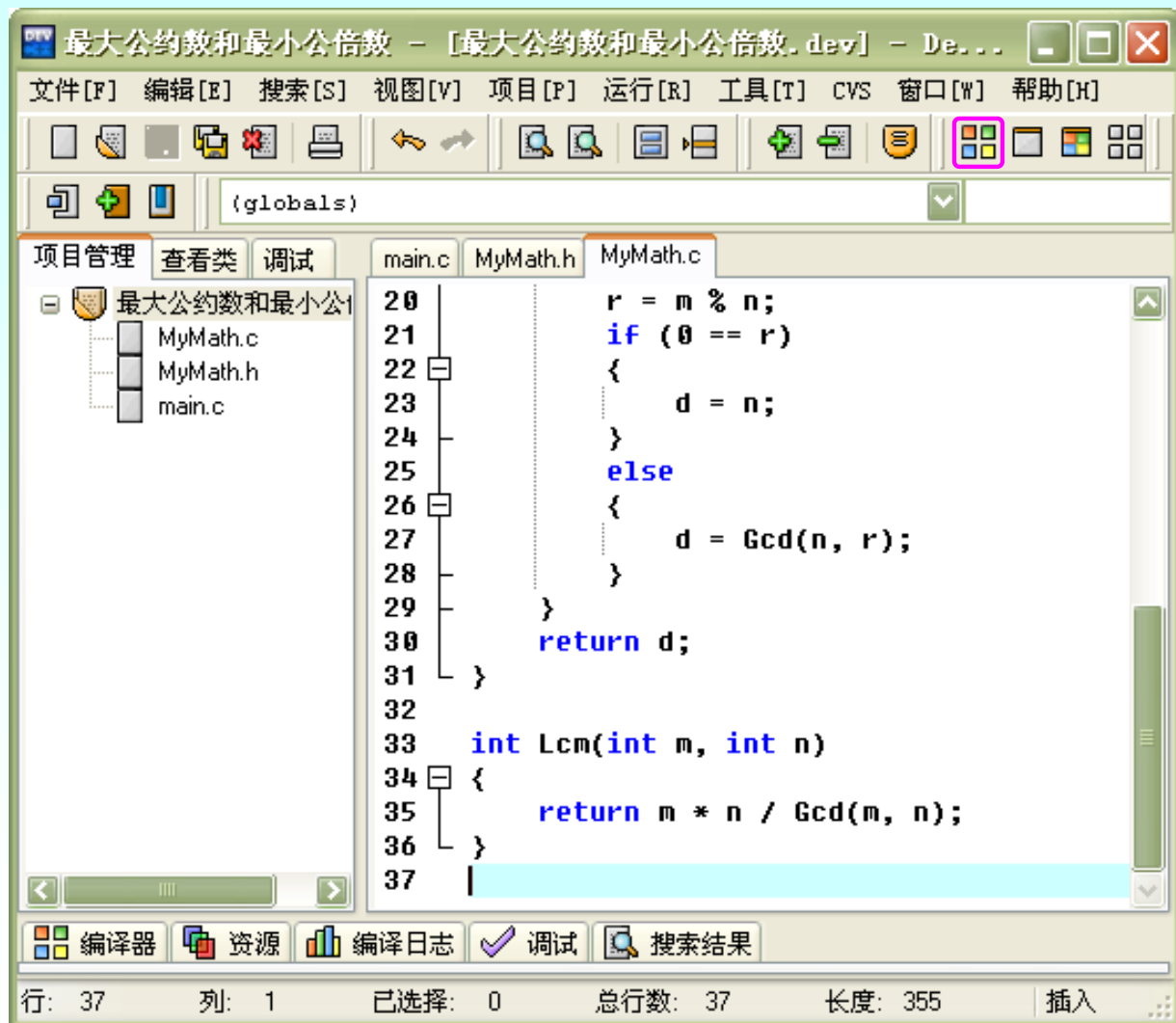
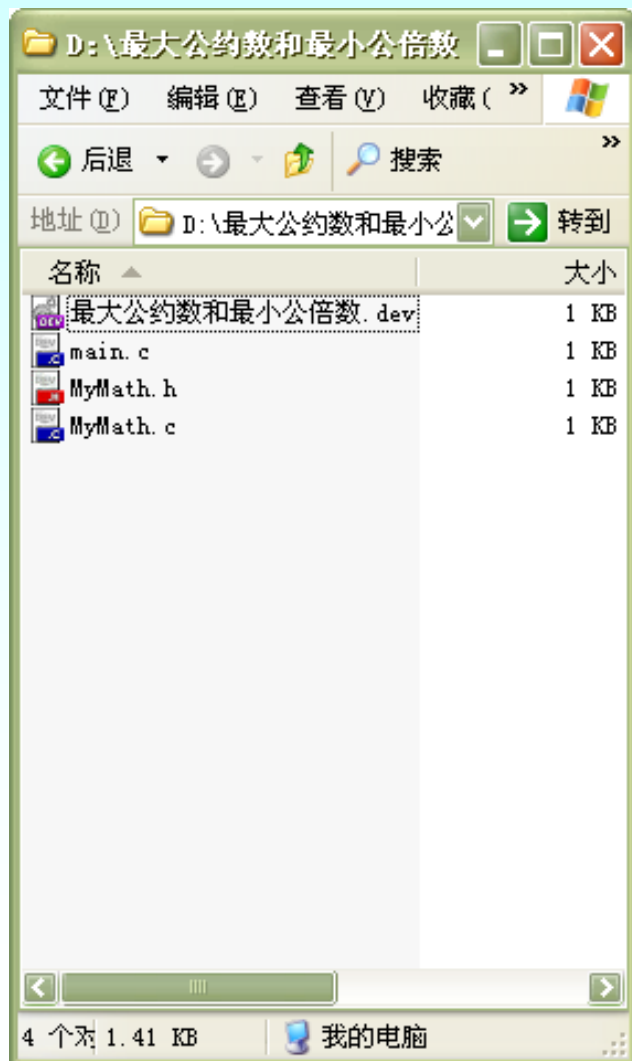


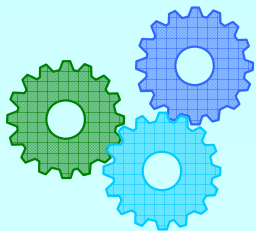
保存“我的数学”程序文件



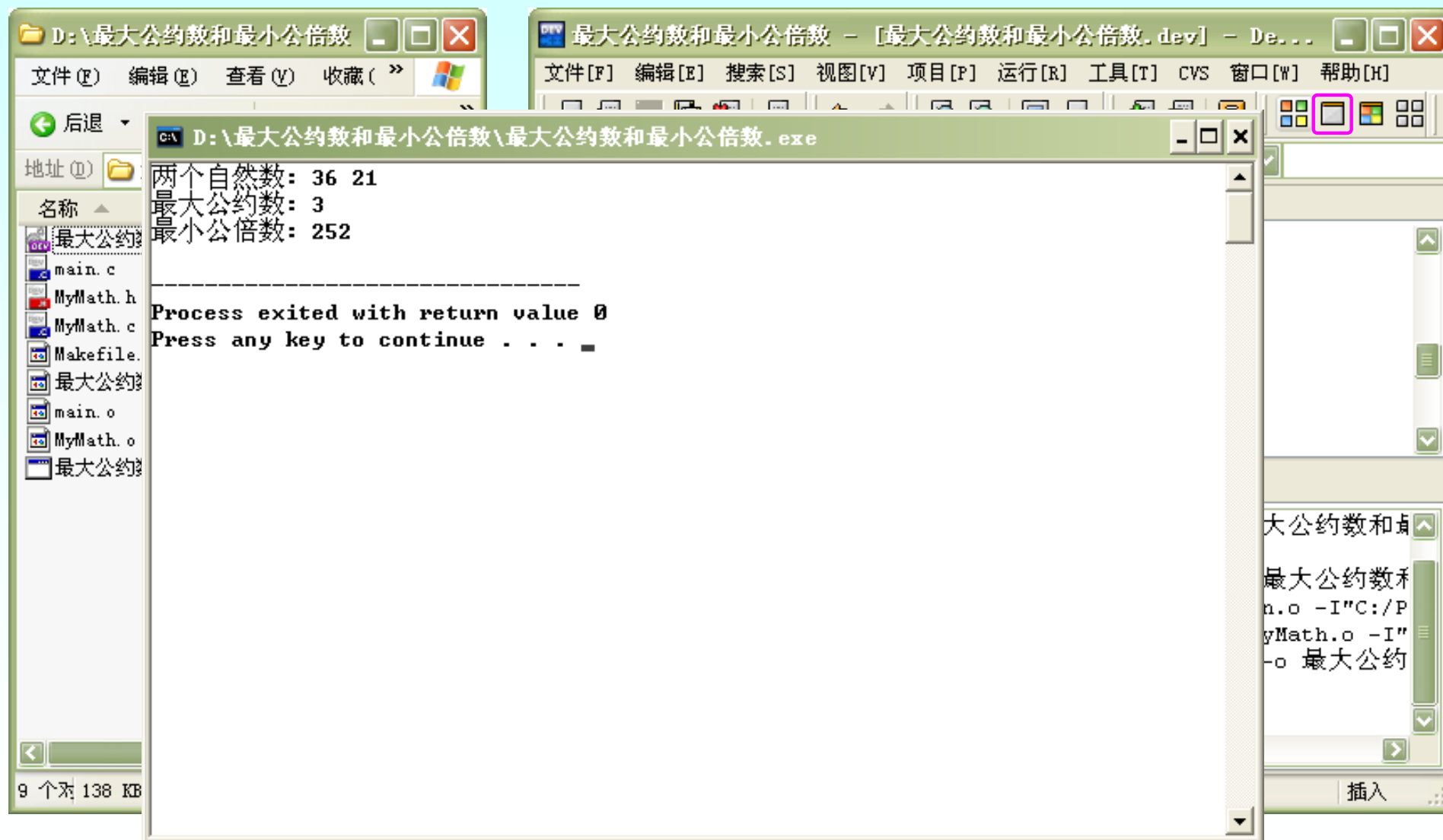


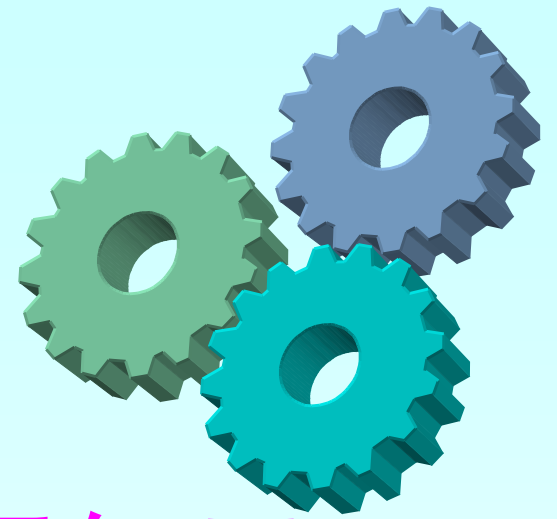
编译和连接





运行





自知者不怨人，知命者不怨天；
怨人者穷，怨天者无志。

——荀子



01100100101101001

自己了解自己的人不抱怨别人，掌握自己命运的人不抱怨老天；抱怨别人的人则穷迫不得志，抱怨老天的人则没有上进的决心和勇气。一个人在工作或事业上尚未获得成功时，不要怨天尤人，要从自身上寻找原因，努力去奋斗。