



中国科学技术大学  
University of Science and Technology of China

# C语言程序调试&习题讲解



# C语言程序调试

- 什么是调试？
  - 在C语言编程过程中，避免不了地会出现各种错误，我们称之为：BUG；
  - 消除这些BUG，我们称之为：Debug；
  - DEBUG中文翻译为：调试。



# C语言程序调试

- C语言有几种BUG?
  - 语法错误：
    - 由于编写的代码不符合C语言语法规范造成的错误；
    - 出现语法错误的程序不能编译、连接生成可执行文件（EXE），所以不能运行；
  - 逻辑错误：
    - 程序已进入到运行状态；
    - 程序运行的结果不对；
    - 在逻辑上程序出现了混乱。



# C语言程序调试

- 语法错误调试方法：
  - 注意信息窗口中**错误**与**警告**的数量；
  - **先**消除**错误****后**消除**警告**；
  - 消除错误的过程中一定要有先后**次序**
  - **前面**的错误没解决之前不要去消除**后面**的！！



# C语言程序调试

- 逻辑错误调试方法—过程：
  - 发现错误结果；
  - 断定显示错误结果对应的变量；
  - 添加并监视该变量；
  - 在执行该变量的前一行设置断点；
  - 执行程序等待程序停下来；
  - 判断出错原因。



# 常见错误提示（警告类）

- **'XXX' declare but never used** 变量XXX已定义但从未用过。
- **'XXX'is assigned a value which is never used** 变量XXX已赋值但从未用过。
- **Code has no effect** 程序中含有没有实际作用的代码。
- **Non-portable pointer conversion** 不适当的指针转换，可能是在应该使用指针的地方用了一个非0的数值。
- **Possible use of 'XXX'before definition** 表达式中使用了未赋值的变量
- **Redeclaration of 'main'** 一个程序文件中主函数main不止一个。
- **Suspicious pointer conversion** 可疑的指针转换。通常是使用了基本类型不匹配的指针。
- **Unreachable code** 程序含有不能执行到的代码。



# 常见错误提示（错误一）

- **Compound statement missing }** in function main 程序结尾缺少括号}。
- **“}”expected;** **“(”expected**等 复合语句或数组初始化的结尾缺少“)”; “(”。
- **Case outside of switch case** 不属于Switch结构，多由于switch结构中的花括号不配对所致。
- **Case statement missing ‘:’** switch结构中的某个case之后缺少冒号。
- **Constant expression required** 定义数组时指定的数组长度不是常量表达式。
- **Declaration syntax error** 结构体或联合类型的定义后缺少分号。
- **Declaration was expected** 缺少说明，通常是因为缺少分界符如逗号、分号、右圆括号等所引起的。
- **Default outside switch Default**部分放到了switch结构之外，一般是因为花括号不匹配而引起的。
- **do statement must have while** do语句中缺少相应的while部分。
- **Expression syntax** 表达式语法错。如表达式中含有两个连续的运算符
- **Extra parameter in call ‘fun’** 调用函数fun时给出了多余的实参。



# 常见错误提示（错误二）

- **Function should return a value** 函数应该返回一个值，否则与定义时的说明类型不匹配。
- **Illegal use of pointer** 指针被非法引用，一般是使用了非法的指针运算。
- **Invalid pointer addition** 指针相加非法。一个指针（地址）可以和一个整数相加，但两个指针不能相加。
- **Lvalue required** 赋值运算的左边是不能寻址的表达式。
- **Misplaced else** 程序遇到了没有配对的else
- **No matching** 表达式中的括号不配对。
- **Pointer required on left side of\_>** 在“\_>”运算的左边只能允许一个指针而不能是一个一般的结构体变量或联合类型的变量。
- **Statement missing;** 程序遇到了后面没有分号的语句。
- **Too few parameters in call** 调用某个函数时实参数目不够。
- **Unable to open include file 'XXXXXXXXX. XXX'** 头文件找不到。
- **Unexpected }或:或{** 在不希望的地方使用了}或: {。
- **Undefined symbol 'X'in function fun** 函数fun中的变量X没有定义。



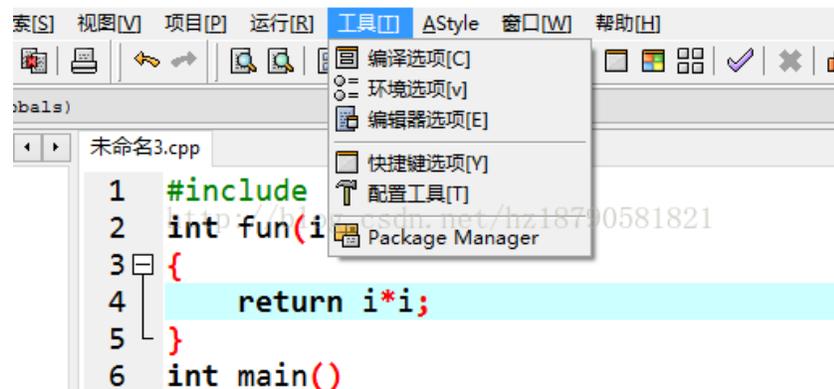
# 常见错误提示（其它）

- 连接中的常见错误
  - 主要错误类似于“undefined symbol `_print` in modula xxx”(print没有定义), 通常是函数名书写错误。
- 运行中的常见错误
  - Abnormal program termination 程序异常终止。通常是由于内存使用不当所致。
  - Floating point error : Domain 或 Divide by 0 运算结果不是一个数或被0除
  - Null pointer assignment 对未初始化的指针赋值, 程序有严重错误。
  - User break 在运行程序时终止

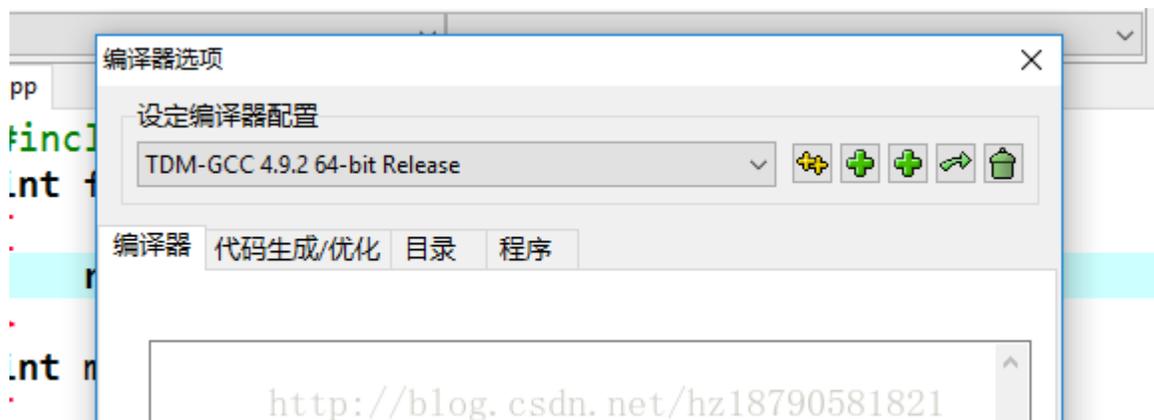


# DevC++调试模式设置

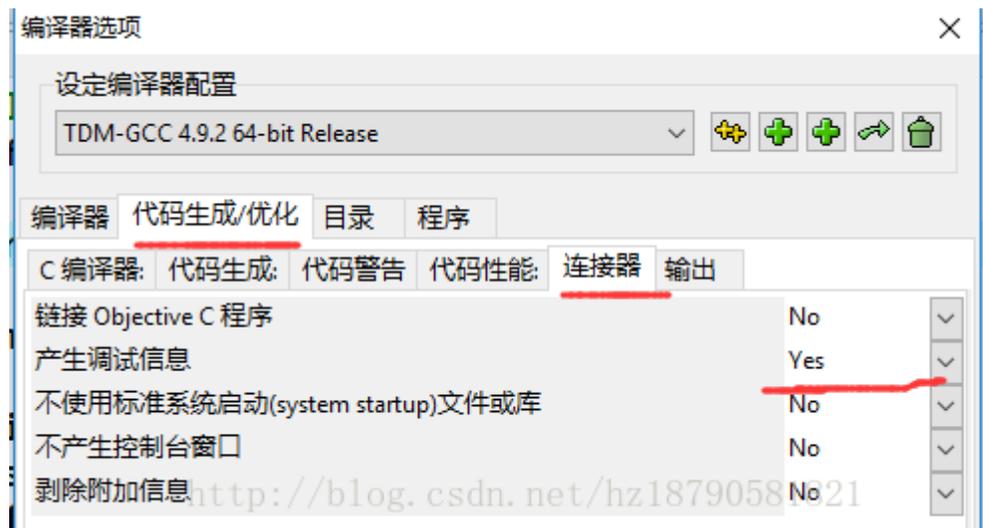
## 1、点开工具



## 2、选择编译器选项



## 3、更改产生调试信息为yes





# DevC++调试方法

## 1、添加断点

```
1 // 编译选项: g++ 1-10-1.c -o 1-10-1.exe
2 int fun(int i)
3 {
4     return i*i;
5 }
6 int main()
7 {
8     int n,m,sum=0;
9     scanf("%d",&n);
10    for(int i=1;i<=n;i++)
11    {
12        sum=sum+fun(i);
13    }
14    printf("%d\n",sum);
15    return 0;
16 }
```

## 2、开始调试

```
3 {
4     return i*i;
5 }
6 int main()
7 {
8     int n,m,sum=0;
9     scanf("%d",&n);
10    for(int i=1;i<=n;i++)
11    {
12        sum=sum+fun(i);
13    }
14    printf("%d\n",sum);
15    return 0;
16 }
```

编译器 资源 编译日志 调试 搜索结果 关闭

调试[D]    添加查看[A]    下一步[N]    跳过[S]    下一条语句

停止执行     查看CPU窗口[V]     单步进入[B]     跳过函数     进入语句

## 3、如下表示调试已经开始

```
2 int fun(int i)
3 {
4     return i*i;
5 }
6 int main()
7 {
8     int n,m,sum=0;
9     scanf("%d",&n);
10    for(int i=1;i<=n;i++)
11    {
12        sum=sum+fun(i);
13    }
14    printf("%d\n",sum);
15    return 0;
16 }
```



# DevC++调试方法

## 4、添加要查看的变量

```
1 #include <stdio.h>
2 int fun(int i)
3 {
4     return i*i;
5 }
6 int main()
7 {
8     int n,m,sum=0;
9     scanf("%d",&n);
10    for(int i=1;i<=n;i++)
11    {
12        sum=sum+fun(i);
13    }
14    printf("%d\n",sum);
15    return 0;
16 }
```

新变量

输入变量名:

sum

资源 编译日志 调试 搜索结果 关闭

添加查看[A] 下一步[N] 跳过[S] 下一条语句 发送命令到GDB: display i

查看CPU窗口[V] 单步进入[I] 跳过函数 进入语句 (gdb) ->-> prompt



# DevC++调试方法

## 5、观察变量如何变化

```
1  #include <stdio.h>
2  int fun(int i)
3  {
4      return i*i;
5  }
6  int main()
7  {
8      int n,m,sum=0;
9      scanf("%d",&n);
10     for(int i=1;i<=n;i++)
11     {
12         sum=sum+fun(i);
13     }
14     printf("%d\n",sum);
15     return 0;
16 }
```

快捷键:

F5 开始调试

F7 单步调试 (运行下一步)

F8 单步进入函数调试

F9 停止调试

# 调试程序示例1:



计算分段函数:

$$f(x) = \begin{cases} 1/x & x \neq 0 \\ 0 & x = 0 \end{cases}$$

```
#include<stdio.h>
void main()
{
    float x,y;
    printf("input x:\n");
    scanf("%f",x)
    if(x!=0)
        y=1/x;
    else
        y=0;

    printf("f(%.2f)=%.1f\n",x,y);
}
```



```
1 #include<stdio.h>
2 void main()
3 {
4     float x,y;
5     printf("input x:\n");
6     scanf("%f",x)
7     if(x!=0)
8         y=1/x;
9     else
10        y=0;
11
12     printf("f(%.2f)=%.1f\n",x,y);
13 }
14
```

编译器 (4) 资源 编译日志 调试 搜索结果 关闭

行	列	单元	信息
2	11	F:\Dev-Cpp\Mystudy\1029.cpp	[Error] '::main' must return 'int'
		F:\Dev-Cpp\Mystudy\1029.cpp	In function 'int main()':
7	4	F:\Dev-Cpp\Mystudy\1029.cpp	[Error] expected ';' before 'if'
9	4	F:\Dev-Cpp\Mystudy\1029.cpp	[Error] 'else' without a previous 'if'

## 调试程序示例2:



## 计算累加和:

```
#include<stdio.h>
void main()
{
    int i,sum;
    for(i=1;i<=100;i++)
        sum=sum+i;

    printf("sum=%d\n",sum);
}
```



## 调试程序示例2:

```
1  #include<stdio.h>
2  int main()
3  {
4      int i,sum;
5      for(i=1;i<=100;i++)
6          sum=sum+i;
7
8      printf("sum=%d\n",sum);
9      return 0;
10 }
11
12
```

F:\Dev-Cpp\Mystudy\1029.exe

sum=5051

-----  
Process exited after 0.01874 seconds with return value 0

资源



请按任意键继续. . .

# 调试程序示例2:



项目管理 查看类 调试 1029.cpp

sum = 1

```
1 #include<stdio.h>
2 int main()
3 {
4     int i,sum;
5     for(i=1;i<=100;i++)
6         sum=sum+i;
7
8     printf("sum=%d\n",sum);
9     return 0;
10 }
11
```

编译器 资源 编译日志 调试 搜索结果 关闭

调试[D] 添加查看[A] 下一步[N] 跳过[S] 下一条语句

停止执行 查看CPU窗口[V] 单步进入[I] 跳过函数 进入语句

评估: [ ]

发送命令到GDB: display sum

->-> display-end  
->-> pre-prompt  
(gdb)  
->-> prompt



## 调试程序示例3:

### 第5章课后习题第9题:

求1000以内的所有“完数”

```
#include<stdio.h>
int main()
{
    int i,j,sum=0;
    for(i=1;i<=1000;i++){
        for(j = 1;j<=i;j++){
            if(i%j==0){
                sum+=j;
            }
        }
        if(sum==i){
            printf("%d是一个完数\n",i);
            //接下来打印它的因子
            // ...
        }
    }
    return 0;
}
```

# 调试程序示例3:



## 运行结果:

```
F:\Dev-Cpp\Mystudy\1029.exe
1是一个完数
-----
Process exited after 0.03674 seconds with return value 0
请按任意键继续. . .
```

```
i = 0
j = 1
sum = 0
1 #include<stdio.h>
2 int main()
3 {
4     int i,j,sum=0;
5     for(i=1;i<=1000;i++){
6         for(j = 1;j<=i;j++){
7             if(i%j==0){
8                 sum+=j;
9             }
10        }
11        if(sum==i){
12            printf("%d是一个完数\n",i);
13            //接下来打印它的因子
14            // ...
15        }
16    }
17    return 0;
18 }
```

```
i = 2
j = 2
sum = 4
1 #include<stdio.h>
2 int main()
3 {
4     int i,j,sum=0;
5     for(i=1;i<=1000;i++){
6         for(j = 1;j<=i;j++){
7             if(i%j==0){
8                 sum+=j;
9             }
10        }
11        if(sum==i){
12            printf("%d是一个完数\n",i);
13            //接下来打印它的因子
14            // ...
15        }
16    }
17    return 0;
18 }
```

# 调试程序示例3:



运行结果:

```
F:\Dev-Cpp\Mystudy\1029.exe
1是一个完数
-----
Process exited after 0.03674 seconds with return value 0
请按任意键继续. . .
```

*i = 2*  
*j = 1*  
*sum = 0*

```
1 #include<stdio.h>
2 int main()
3 {
4     int i,j,sum=0;
5     for(i=1;i<=1000;i++){
6         sum=0;
7         for(j = 1;j<=i;j++){
8             if(i%j==0){
9                 sum+=j;
10            }
11        }
12        if(sum==i){
13            printf("%d是一个完数\n",i);
14            //接下来打印它的因子
15            // ...
16        }
17    }
18    return 0;
19 }
```

*i = 2*  
*j = 2*  
*sum = 3*

```
1 #include<stdio.h>
2 int main()
3 {
4     int i,j,sum=0;
5     for(i=1;i<=1000;i++){
6         sum=0;
7         for(j = 1;j<=i;j++){
8             if(i%j==0){
9                 sum+=j;
10            }
11        }
12        if(sum==i){
13            printf("%d是一个完数\n",i);
14            //接下来打印它的因子
15            // ...
16        }
17    }
18    return 0;
19 }
```



# 总结

- 编译型错误：
  - 检查语法
  - 报错信息
- 运行时错误：
  - 报错信息
  - debug



中国科学技术大学  
University of Science and Technology of China

# 习题讲解

# 3-3



```
#include<stdio.h>
#include <math.h>

int main()
{
    float d = 300000, p = 6000, r = 0.01, m;
    m = log10(p / (p - d*r)) / log10(1 + r);
    printf("m = %3.1f\n", m);
    return 0;
}
```

# 3-6

```
#include<stdio.h>
```

```
int main()
```

```
{
```

```
    char c1 = 'c', c2 = 'h', c3 = 'i', c4 = 'n', c5 = 'a';
```

```
    c1 = c1 + 4;
```

```
    c2 = c2 + 4;
```

```
    c3 = c3 + 4;
```

```
    c4 = c4 + 4;
```

```
    c5 = c5 + 4;
```

```
    //使用putchar输出
```

```
    printf("使用putchar输出: ");
```

```
    putchar(c1);
```

```
    putchar(c2);
```

```
    putchar(c3);
```

```
    putchar(c4);
```

```
    putchar(c5);
```

```
    printf("\n");
```

```
    //使用printf输出
```

```
    printf("使用putchar输出: %c%c%c%c%c\n", c1, c2, c3, c4, c5);
```

```
    return 0;
```

```
}
```



## 3-7



```
#include<stdio.h>

int main()
{
    float h, r, l, s, sq, vq, vz;
    float pi = 3.141526;
    printf("请输入圆半径r, 圆柱高h:");
    scanf("%f %f", &r, &h); //要求输入圆半径r和圆柱高h
    l = 2 * pi*r; //计算圆周长l
    s = r*r*pi; //计算圆面积s
    sq = 4 * pi*r*r; //计算圆球表面积sq
    vq = 3.0 / 4.0*pi*r*r*r; //计算圆球体积vq
    vz = pi*r*r*h; //计算圆柱体积vz
    printf("圆周长为:          l=%6.2f\n", l);
    printf("圆面积为:          s=%6.2f\n", s);
    printf("圆球表面积为:      sq=%6.2f\n", sq);
    printf("圆球体积为:          v=%6.2f\n", vq);
    printf("圆柱体积为:          vz=%6.2f\n", vz);
    return 0;
}
```

# 4-6



```
#include <stdio.h>
int main()
{
    int x, y;
    scanf_s("%d", &x);
    if (x < 1) {
        y = x;
    }
    else if (x >= 1 && x < 10) {
        y = 2 * x - 1;
    }
    else {
        y = 3 * x - 11;
    }
    printf("y = %d\n", y);
    system("pause");//这一句是为了让控制台不退出
    return 0;
}
```

# 4-8

```
#include <stdio.h>
int main()
{
    int score;
    printf("enter score:");
    scanf_s("%d", &score);
    if (score >= 90) {
        printf("A\n");
    }else if (score >= 80 && score < 90) {
        printf("B\n");
    }else if (score >= 70 && score < 80) {
        printf("C\n");
    }else if (score >= 60 && score < 70) {
        printf("D\n");
    }else {
        printf("E\n");
    }
    system("pause");
    return 0;
}
```



# 4-9 (1)

```
#include <stdio.h>
int main()
{
    int num;
    printf("enter num:");
    scanf_s("%d", &num);
    if (num > 99999 || num < 0) {
        printf("请输入0~99999之间的正数\n");
        return -1;
    }
    if (num >= 10000) {
        printf("5\n");
    }else if (num >= 1000) {
        printf("4\n");
    }else if (num >= 100) {
        printf("3\n");
    }else if (num >= 10) {
        printf("2\n");
    }else {
        printf("1\n");
    }
    system("pause");
    return 0;
}
```



## 4-9 (2)

```
#include <stdio.h>
int main()
{
    int num;
    printf("enter num:");
    scanf_s("%d", &num);
    if (num > 99999 || num < 0) {
        printf("请输入0~99999之间的数字\n");
        return -1;
    }
    if (num / 10000 > 0) { //取出万位数字
        printf("%d ", num / 10000);
    }
    if (num%10000 >= 1000) { //取余10000则可以取出低四位的数据, 除以1000则得到千位的数字
        printf("%d ", (num % 10000) / 1000);
    }
    if (num%1000 >= 100) { //取余1000则可以取出低三位的数据, 除以100则得到百位的数字
        printf("%d ", (num % 1000) / 100);
    }
    if (num%100 >= 10) { //取余100则可以取出低两位的数据, 除以10则得到十位的数字
        printf("%d ", (num % 100) / 10);
    }
    if (num%10 >= 0) { //取余10则取出个位数字
        printf("%d ", num % 10);
    }
    printf("\n");
    system("pause");
    return 0;
}
```



# 4-9 (3)

```
#include <stdio.h>
int main()
{
    int num;
    printf("enter num:");
    scanf_s("%d", &num);
    if (num > 99999 || num < 0) {
        printf("请输入0~99999之间的数字\n");
        return -1;
    }
    if (num % 10 >= 0) {
        printf("%d ", num % 10);
    }
    if (num % 100 >= 10) {
        printf("%d ", (num % 100) / 10);
    }
    if (num % 1000 >= 100) {
        printf("%d ", (num % 1000) / 100);
    }
    if (num % 10000 >= 1000) {
        printf("%d ", (num % 10000) / 1000);
    }
    if (num / 10000 > 0) {
        printf("%d ", num / 10000);
    }
    printf("\n");
    system("pause");
    return 0;
}
```



# 4-10 (1)

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    double I, salary = 0;
```

```
    printf("enter performance:");
```

```
    scanf_s("%lf", &I);
```

```
    if (I < 0) {
```

```
        printf("请输入一个正数\n");
```

```
        system("pause");
```

```
        return -1;
```

```
    }
```

```
    double salary1 = 100000 * 0.1; //10万的奖金
```

```
    double salary2 = (200000 - 100000) * 0.075 + salary1; //20万的奖金
```

```
    double salary3 = (400000 - 200000) * 0.05 + salary2; //40万的奖金
```

```
    double salary4 = (600000 - 400000) * 0.03 + salary3; //60万的奖金
```

```
    double salary5 = (1000000 - 600000) * 0.015 + salary4; //100万的奖金
```

```
    if (I <= 100000) {
```

```
        salary = I * 0.1; //小于100000按10%提成
```

```
    }else if (I > 100000 && I <= 200000) {
```

```
        salary = salary1 + (I - 100000) * 0.075; //多出10万的按比例计算, 加上10w的奖金
```

```
    }else if (I > 200000 && I <= 400000) {
```

```
        salary = salary2 + (I - 200000) * 0.05; //多出20万的按比例计算, 加上20w的奖金
```

```
    }else if (I > 400000 && I <= 600000) {
```

```
        salary = salary3 + (I - 400000) * 0.03; //多出40万的按比例计算, 加上40w的奖金
```

```
    }else if (I > 600000 && I <= 1000000) {
```

```
        salary = salary4 + (I - 600000) * 0.015; //多出60万的按比例计算, 加上60w的奖金
```

```
    }else if (I > 1000000){
```

```
        salary = salary5 + (I - 1000000) * 0.01; //多出100万的按比例计算, 加上100w的奖金
```

```
    }
```

```
    printf("salary:%f\n", salary);
```

```
    system("pause");
```

```
    return 0;
```

```
}
```



# 4-10 (2)

```
#include <stdio.h>
int main()
{
    double I, salary = 0;
    printf("enter performance:");
    scanf_s("%lf", &I);
    if (I < 0) {
        printf("请输入一个正数\n");
        system("pause");
        return -1;
    }
    double salary1 = 100000 * 0.1; //大于100000时0~100000的奖金
    double salary2 = (200000 - 100000) * 0.075 + salary1; //大于200000时0~20万的奖金
    double salary3 = (400000 - 200000) * 0.05 + salary2; //大于400000时0~40万的奖金
    double salary4 = (600000 - 400000) * 0.03 + salary3; //大于600000时0~60万的奖金
    double salary5 = (1000000 - 600000) * 0.015 + salary4; //大于1000000时0~100万的奖金
    int grade = I / 100000;
    switch(grade) {
        case 0:
            salary = I * 0.1; break;
        case 1:
            salary = salary1 + (I - 100000) * 0.075; break;
        case 2: //会顺序执行到下一个break处
        case 3:
            salary = salary2 + (I - 200000) * 0.05; break;
        case 4:
        case 5:
            salary = salary3 + (I - 400000) * 0.03; break;
        case 6:
        case 7:
        case 8:
        case 9:
            salary = salary4 + (I - 600000) * 0.015; break;
        default:
            salary = salary5 + (I - 1000000) * 0.01; break;
    }
    printf("salary:%f\n", salary);
    system("pause");
    return 0;
}
```



# 5-3

```
#include <stdio.h>
int main()
{
    int p, r, n, m, temp;
    printf("请输入两个正整数n,m:");
    scanf("%d%d", &n, &m);
    //调整n保存较大的值
    if (n < m)
    {
        temp = n;
        n = m;
        m = temp;
    }

    p = n * m;
    while (m != 0)
    {
        r = n % m;
        n = m;
        m = r;
    }
    printf("它们的最大公约数为:%d\n", n);
    printf("它们的最小公倍数为:%d\n", p / n);
    return 0;
}
```



# 5-4

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    char c;
```

```
    //定义eng_char为英文字母的个数, 初始值为0
```

```
    //定义space_char为空格字符的个数, 初始值为0
```

```
    //定义digit_char为数字字符的个数, 初始值为0
```

```
    //定义other_char为其他字符的个数, 初始值为0
```

```
    int eng_char = 0, space_char = 0, digit_char = 0, other_char = 0;
```

```
    printf("请输入一行字符: ");
```

```
    while ((c = getchar()) != '\n')
```

```
    {
```

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

```
        {
```

```
            eng_char++;
```

```
        }
```

```
        else if (c == ' ')
```

```
        {
```

```
            space_char++;
```

```
        }
```

```
        else if (c >= '0' && c <= '9')
```

```
        {
```

```
            digit_char++;
```

```
        }
```

```
        else
```

```
        {
```

```
            other_char++;
```

```
        }
```

```
    }
```

```
    printf("英文字母数量:%d\n空格数量:%d\n数字数量:%d\n其他字符数量:%d\n", eng_char, space_char, digit_char, other_char);
```

```
    return 0;
```

```
}
```





```
#include <stdio.h>

int main()
{
    //a表示百位数字, b表示十位数字, c表示各位数字
    int a, b, c;
    for (int i = 100; i <= 999; i++)
    {
        a = i / 100;
        b = (i / 10) % 10;
        c = i % 10;
        if (a * a * a + b * b * b + c * c * c == i)
        {
            printf("%d\n", i);
        }
    }
    return 0;
}
```

# 5-9

```
#include<stdio.h>
```

```
int main()
```

```
{
```

```
    int data, fator, sum;      /* data表示要判断的数, fator表示因子, sum表示因子之和*/
```

```
    for (data = 2; data <= 1000; data++)
```

```
    {
```

```
        //1是所有整数的因子, 所以因子之和从1开始
```

```
        sum = 1;
```

```
        for (fator = 2; fator <= data / 2; fator++)
```

```
        {
```

```
            /* 判断data能否被fator整除, 能的话fator即为因子 因子不包括自身 */
```

```
            if (data % fator == 0)
```

```
            {
```

```
                sum += fator;
```

```
            }
```

```
        }
```

```
        // 判断此数是否等于因子之和 */
```

```
        if (sum == data)
```

```
        {
```

```
            printf("%d its factors are 1, ", data);
```

```
            for (fator = 2; fator <= data / 2; fator++)
```

```
            {
```

```
                if (data % fator == 0)
```

```
                {
```

```
                    printf("%d, ", fator);
```

```
                }
```

```
            }
```

```
            printf("\n");
```

```
        }
```

```
    }
```

```
    return 0;
```

```
}
```



# 5-12



```
#include <stdio.h>

int main()
{
    int day = 9;
    int prev_day_count;
    int cur_day_count = 1;
    while (day > 0)
    {
        prev_day_count = (cur_day_count + 1) * 2;
        cur_day_count = prev_day_count;
        day--;
    }
    printf("total count : %d\n", cur_day_count);
    return 0;
}
```