



# 第一讲 程序设计入门

Lv Jiuyang

# 目录

## 第零节 熟悉集成开发环境

## 第一节 第一个程序：Hello, world!

## 第二节 C语言基本语法

- 变量和常量
- 赋值语句
- 基本数据类型
- 标准输入输出
- 运算符和表达式
- 常用数学函数
- 常见错误

## 第三节 顺序结构与选择结构

# 第零节 熟悉集成开发环境

# 程序运行的原理

C语言代码 -----> 机器可执行的二进制文件 -----> 可执行程序  
(人类可读) (机器可读)

因此，书写高级语言必须遵循一些语法规则，机器才能理解人的指令。

# 算法竞赛一般使用的IDE

- Dev-C++（推荐）
- Visual Studio
- CodeBlocks

# 算法竞赛一般使用的文本编辑器

（需要自己配置环境）

- VScode
- Clion
- Notepad++
- Vim

# 第一节 第一个程序：Hello, world!

# Hello, world!

编写一个能够输出“Hello, world!”的C语言程序。

```
#include <stdio.h>

int main()
{
    printf("Hello, world!");
    return 0;
}
```

```
#include <stdio.h> // 头文件

int main() // 主函数
{
    printf("Hello, world!"); // 调用 printf 函数
    return 0; // 返回0，主函数中可不写，主函数不能返回其他值
}
```

- 头文件主要用于保存程序的声明
- 一个C语言程序有且仅有一个 `main` 函数，任何一个C程序总是从 `main` 函数开始执行
- `printf` 函数是 `<stdio.h>` 头文件中包含的一个函数



## 第二节 C语言基本语法

## 2.1 变量和常量

### 变量

1. 变量名是字母、数字和下划线的组合，开头必须是字母或下划线，不能是数字。
2. 变量名不可以是C语言关键字
3. C语言变量区分大小写

### 常量

`const` 常量定义后无法对值更改

- 尽量不要用宏定义

```
#define x 10  
int MaxSpeed;  
const int maxn = 1e5 + 5;
```

## 2.2 赋值语句

格式：

变量 = 表达式；

意义：

将右边表达式的值赋给左边的变量

- 一定要区分 `=` 和 等于，C语言中判断相等用 `==` 运算符
- 在进行赋值运算时，如果赋值运算符两边的数据类型不同，系统将会自动进行类型转换，即将赋值运算符右边的数据类型转换成左边的变量类型。可能会出现数据丢失。

```
int a = 0x7fffffff;  
int b = a;  
a = a - b;
```

## 2.3 基本数据类型

### 2.3.1 整型（integer）

数据类型	定义标识符	占字节数	数值范围
短整型	<code>short</code>	2	$-2^{15} \sim 2^{15}-1$ (32767)
整型	<code>int</code>	4	$-2^{15} \sim 2^{15}-1$ (约 $2 \times 10^9$ )
超长整型	<code>long long</code>	8	$-2^{63} \sim 2^{63}-1$ (约 $9 \times 10^{18}$ )
无符号整型	<code>unsigned int</code>	4	$0 \sim 2^{32}-1$ (约 $4 \times 10^9$ )

其余无符号类型同理

```
int a;  
unsigned long long b = 1;
```

## 2.3.2 实型

数据类型	定义标识符	数值范围	占字节数	有效位数
单精度实型	float	$-3.4 \times 10^{38} \sim 3.4 \times 10^{38}$	4	6~7位
双精度实型	double	$-1.7 \times 10^{308} \sim 1.7 \times 10^{308}$	8	15~16位
长双精度实型	long double	$-3.4 \times 10^{4932} \sim 1.1 \times 10^{4932}$	16	18~19位
布尔型	bool	true 或 false	1	

```
const double pi = 3.1415926;  
bool flag1 = true, flag2 = 1;
```

## 2.3.3 字符型

定义标识符	占字节数	数值范围
char	1	-128~127

### 1. 直接表示

```
char c = 'A';
```

### 2. ASCII码表示

```
char ch = 65;
```

### 3. 转义字符

换行符 '\n'

单引号 '\''

反斜杠 '\\'

- 字符型也可以做加减运算

## 2.4 标准输入输出

头文件: `<stdio.h>`

输入函数:

```
scanf("%对应格式字符", &变量名);
```

输出函数:

```
printf("其他字符 %对应格式字符 其他字符", 变量名);
```

- 一定不要忘记 `scanf` 函数中的`&`
- 不要使用 `<conio.h>` 头文件的函数, 如 `getch()` 等
- 字符类型变量名 = `getchar()`; 使用时注意吃掉换行符

数据类型	格式符*	数据类型	格式符*
int	%d	long long	%lld
unsigned long long	%llu		
char	%c	字符数组	%s
float	%f	double	输入 %lf 输出 %f

\*这里为了看着方便加上了%

```
int a;
long long b;
char c;
double d = 1.0;
float f = 2.7;

scanf("%d%lld", &a, &b);
scanf("%c", &c); // 空格, 回车, 缩进都会被读入, 注意!
printf("%d%d", a, c); // c被转换成int类型
printf("ans = %f %.12f\n", f, d); // 精确到小数点后12位
```



## 2.5 运算符和表达式

### 2.5.1 运算符和表达式

运算符类型	常用运算符	示例
算术运算符	<code>+</code> <code>-</code> <code>*</code> <code>/</code> <code>%</code> <code>++</code> <code>--</code>	<code>5 % 2</code>
关系运算符	<code>&gt;</code> <code>&lt;</code> <code>==</code> <code>&gt;=</code> <code>&lt;=</code> <code>!=</code>	<code>a == 1</code>
逻辑运算符	<code>&amp;&amp;</code> <code>  </code> <code>!</code>	<code>a&gt;0 &amp;&amp; a&lt;b</code>
位操作运算符	<code>&amp;</code> <code> </code> <code>~</code> <code>^</code> <code>&lt;&lt;</code> <code>&gt;&gt;</code>	<code>1&lt;&lt;2</code>
赋值运算符	<code>=</code> <code>+=</code> <code>&amp;=</code> <code>&lt;&lt;=</code>	<code>a += 2</code>
条件运算符	<code>?</code> <code>:</code>	<code>a == 1 ? 1 : 0</code>
指针运算符	<code>*</code> <code>&amp;</code>	<code>*it</code>

## 2.5.2 运算符优先级

运算符的优先级如下：

赋值运算符 < 逻辑运算符 < 关系运算符 < 算术运算符

- 不确定就加括号

```
int a = 1, b = 2, c, d;  
a = a != b ? b : a; // 不要写这种没意义的代码  
c = a > b;  
d = a = b;  
d = a == b;  
(d = a) == b;
```

```
int a = 2, b = 2, ans;  
ans = a++; // ans = 2  
ans = ++b; // ans = 3
```

## 2.6 常用数学函数

头文件: `<math.h>`

函数名	格式	功能	示例
绝对值函数	<code>abs(x)</code>	求一个数 $x$ 的绝对值	<code>abs(-5)</code>
向上取整	<code>ceil(x)</code>	求不小于实数 $x$ 的最小整数	<code>ceil(3.14)</code>
向下取整	<code>floor(x)</code>	求不大于实数 $x$ 的最大整数	<code>floor(3.14)</code>
自然对数函数	<code>log(x)</code>	求实数 $x$ 的自然数对数	<code>log(1)</code>
指数函数	<code>pow(x,y)</code>	计算 $x^y$	<code>pow(2,3)</code>
平方根值函数	<code>sqrt(x)</code>	求实数 $x$ 的平方根	<code>sqrt(25)</code>



不要打扰我写bug

## 2.7 初学者常犯的错误

```
int mian()  
{  
    int a, b; c, d = 1;  
    bool flag1 = ture, flag2 = flase;  
    scanf("%d", a);  
    printf("%d, b");  
    c = b + 1  
    e = 2a;  
    return 1;  
}
```

以上代码除了花括号所在行，每行至少有一个错误，共11处

## 2.8 注意

以下问题编译器不会报错，但仍然无法得出正确结果

- 整数除法 两个整数相除结果还是整数，`5/2` 的值为2而不是2.5  
方法：至少一个数转换成浮点数，如 `5/2.0` `(float)a/b`
- 数据溢出

```
int a = 100000;  
long long b = a * a; // Wrong Answer  
long long b = 1LL * a * a; // Accepted
```

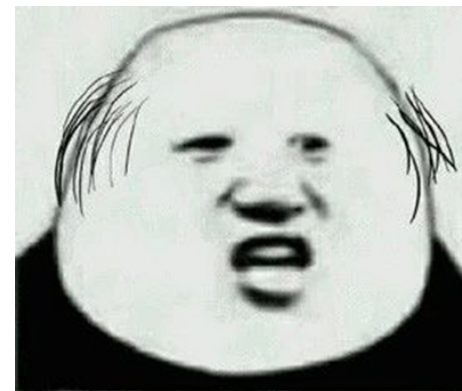
- `ceil`, `floor`, `pow` 函数结果都为double型，  
不能直接与整型判断相等，可能有精度损失

```
double x = 3.14; int a = 3;  
if(floor(3.14) == a) // Wrong Answer  
if(fabs(floor(3.14) - a) <= 1e-6) // Accepted  
...
```

## 第三节 顺序结构与选择结构

# 程序设计基本思路

输入 --> 计算处理 --> 输出



程序，搞起来很轻松的  
就是头冷

# 交换变量

## 问题描述：

传媒大学幼儿园开学了，老师给好学的zls出了一道难题，zls拿着题目找到了ACM队的新生。你能帮帮zls吗？

## 输入：

整数a和整数b，( $0 < a, b < 10^9$ )

## 输出：

交换后a和b的值，中间以一个空格分开

## 样例输入：

3 4

## 样例输出：

4 3



法一：

```
#include <stdio.h>
int main()
{
    int a, b, t;
    scanf("%d%d", &a, &b);
    t = a; a = b; b = t;
    printf("%d %d\n", a, b);
}
```

法二：（以下省略头文件）

```
int main()
{
    int a, b, t;
    scanf("%d%d", &a, &b);
    printf("%d %d\n", b, a);
}
```

法三：

```
int main()
{
    int a, b;
    scanf("%d%d", &a, &b);
    a = a + b;
    b = a - b;
    a = a - b;
    printf("%d %d\n", a, b);
}
```

还有没有其他方法？

- 类似法三的原理 `a = a * b`，两个变量都应使用 `long long`
- 使用 `^` 运算符（同为0，异为1）
- C++中的 `swap` 函数
- 指针

## 3.1 if选择结构

格式:

```
if(条件表达式)  
    语句1;
```

```
if(条件表达式) {  
    语句1;  
    语句2;  
}
```

```
if(条件表达式) {  
    语句1;  
    语句2;  
}  
else if(条件表达式) {  
    语句3;  
}  
else {  
    语句4;  
}
```

## 常见错误

```
if(a = 1) printf("Yes\n");
```

## 程序设计风格提示

从语法上来讲，整个 `if` 语句可以写在一行。如果条件表达式和语句都非常简单，那么整个语句可以写在一行；否则，最好在条件表达式后换行，而且语句部分要相对 `if` 缩进。

三目运算符 `?:` 可以用来代替 `if-else` 结构。

# 空调温度

## 问题描述：

zls的作业得到了老师和同学们的一致好评，连空调都开始吹zls了！zls觉得有点冷，他决定请你写一个程序来控制空调的开关。

## 输入：

室内温度 $t$  ( $10 < t < 40$ )。

## 输出：

当温度低于25度时，输出0表示空调关闭，否则输出1表示空调开启。

## 样例输入：

30

## 样例输出：

1

法一：

```
#include <stdio.h>
int main()
{
    int t;
    scanf("%d", &t);
    if(t < 25)
        printf("0\n");
    else
        printf("1\n");
}
```

法二：（省略头文件）

```
int main()
{
    int t;
    scanf("%d", &t);
    printf("%d\n", t < 25 ? 0 : 1);
}
```

## 3.2 switch语句



格式:

```
switch(表达式)
{
    case 常量表达式1:
        语句序列1;
        break;
    case 常量表达式2:
        语句序列2;
        语句序列3;
        break;
    ...
    default: //可省略
        语句序列n;
}
```

- 不要忘记 `break` 和每种情况后的 `:`
- `switch` 语句的对象只能是 `int` `char` `bool` 类型的数据

# zls学英语

## 问题描述：

zls太难了！英语老师今天教了一周七天的七个单词，zls记了很久也没记住，你能帮帮他吗？

## 输入：

一个整数，表示今天星期几

## 输出：

对应的英文，若数据非法则输出error

## 样例输入：

1

## 样例输出：

Monday

## 样例输入：

8

## 样例输出：

error



```
#include <stdio.h>
int main()
{
    int a;
    scanf("%d", &a);
    switch(a) {
        case 1: printf("Monday\n"); break;
        case 2: printf("Tuesday\n"); break;
        case 3: printf("Wednesday\n"); break;
        case 4: printf("Thursday\n"); break;
        case 5: printf("Friday\n"); break;
        case 6: printf("Saturday\n"); break;
        case 7: printf("Sunday\n"); break;
        default: printf("error\n");
    }
    return 0;
}
```

## 课后练习

# zls的GPA

问题描述：

zls收到了上学期的成绩，他对队友zls(♀)说：我的GPA比你高3分！请问zls的成绩是多少呢？

输入：

zls(♀)的GPA，保证在 $[2, 4]$ 范围内

输出：

zls的GPA，保留小数点后一位

样例输入：

2

样例输出：

5.0

样例输入：

3.4

样例输出：

6.4



zlsnb

# zls判断闰年

## 问题描述：

zls觉得一年好短啊，为什么每年不能像闰年一样有366天呢？

请你编写程序判断今年是否是闰年。（闰年的条件：1.能整除4且不能整除100； 2.能整除400，满足一个条件就是闰年）

## 输入：

一个正整数，代表年份，保证在 `int` 范围内

## 输出：

如果是闰年，则输出yes，否则输出no

## 样例输入：

2020

## 样例输出：

yes

## 样例输入：

1900

## 样例输出：

no

# zls的连续和

## 问题描述：

zls今天看了数学家高斯的故事，zls觉得连续和这么简单的题目他也能做出来。

## 输入：

一个正整数 $n(0 < n \leq 4 * 10^9)$ ，注意数据范围

## 输出：

$1 + 2 + \dots + n$  的值，保证结果在 `long long` 的范围内

## 样例输入：

100

## 样例输出：

5050

## 样例输入：

4000000000

## 样例输出：

80000000002000000000

## Div.2 警告！

### 问题描述：

屡教不改的zls今天又写了很多bug，他很不爽，他决定让lly帮他把语数外成绩从小到大排序，如果做不出来就会给lly布置Div.2，lly不想做Div.2，你能帮帮他吗？

### 输入：

三个正整数 $a, b, c$ , ( $0 < a, b, c \leq 10^9$ )

### 输出：

从小到大排序后的三个整数，中间以一个空格隔开

### 样例输入：

3 5 4

### 样例输出：

3 4 5

### 样例输入：

4 3 5

### 样例输出：

3 4 5

## zls的高考题

### 问题描述：

zls今天看见了这么多师弟师妹来A914学习，不由得兴奋地想起当年的自己。想到这，zls情不自禁地拿出2016全国三卷理科数学高考最后一道选择题准备再体验一把当年秒题的快感。原题如下：

定义“规范01数列”  $a_n$  如下:  $a_n$  共有  $2m$  项, 其中  $m$  项为0,  $m$  项为1, 且对任意  $k \leq 2m$ ,  $a_1, a_2, \dots, a_k$  中0的个数不少于1的个数. 若  $m = 4$ , 则不同的“规范01数列”共有 (14) 个。

当然，zls很快就做出来这道题，但他还是认为这道题太简单了，他在想  $m$  取其他值结果是多少呢？

输入:

一个正整数 $m$ , ( $1 \leq m \leq 8$ )

输出:

问题的解, 保证结果在 `int` 范围内

样例输入:

4

样例输出:

14

样例输入:

8

样例输出:

1430



# zls的24点

## 问题描述:

“24点”是一种益智游戏，玩法是把扑克牌上4个数字通过加减乘除以及括号运算，使最后的计算结果是24。如 $(3+5-4)\times 6=24$ 。

zls觉得一般的24点太无聊了，他发明了“zls的24点”。盒子里装着 $n$ 张写有数字的扑克牌，数字大小不限，可以进行任意次以下操作：

1. 对于盒子中的任意两个数  $a, b$ ，可以进行

$$(1) a + b \quad (2) a - b \quad (3) a \times b \quad (4) a \div b \quad (b \neq 0) \quad (5) a^b \quad (6) \log_a b$$

任何一个操作后并将结果放回盒子，所有运算都遵从数学规律。

2. 对于盒子中的任意一个数  $a$ ，可以进行

$$(1) a! (\text{阶乘}) \quad (2) \frac{a}{100} \quad (3) \sqrt{a} \quad (4) |a| \quad (5) \lceil a \rceil (\text{向上取整}) \quad (6) \lfloor a \rfloor (\text{向下取整})$$

任何一个操作后并将结果放回盒子，所有运算都遵从数学规律。

输入：

游戏开始时扑克牌的数目 $n$  ( $1 \leq n \leq 5$ ),

接下来  $n$  个整数，代表  $n$  张扑克牌上面的数值。

输出：

若进行若干次操作后，盒子中能只剩下24一个数，则输出Yes，反之输出No。

样例输入：

4 0 0 0 0

样例输出：

Yes

样例说明：

$$(0! + 0! + 0! + 0!)! = 24$$

只有一个0时无论如何也计算不出24，故输出No

样例输入：

1 0

样例输出：

No

## zls的24点 测试数据

样例输入：

2 0 2

样例输出：

Yes

样例输入：

4 4 4 4 4

样例输出：

Yes

样例输入：

5 100000 100000 100000 100000 100024

样例输出：

Yes

# 课程预告

- 9.21（周六） 13:30 ~ 17:00 《循环结构程序设计》 zls(♀)
- 9.26（周四） 18:00 ~ 22:00 《分支循环思维练习》 zls