

# 程序设计语言与方法(C语言)

## 第四章 键盘输入与屏幕输出



# C程序结构

```
#include <stdio.h>
#include <math.h>
#define PI 3.14159
```

/\* 预处理命令 \*/

```
int main()
{
```

/\* 主函数，一个程序有且只有一个 \*/

```
    int a, b, c;
    int x = 5;
    double y;
```

/\* 变量定义语句 \*/

```
    y = x * 10;
    y = y + a * b - c;
    printf("%lf\n", sqrt(y));
```

/\* 数据处理语句 \*/

```
    return 0;
}
```

/\* 返回，程序结束 \*/



# 编写程序：求1到10的累加和

- ❖ 分析问题（略）
- ❖ 寻求解决/计算过程（略）
- ❖ 描述计算过程
- ❖ 编写源程序



# 描述问题：1到10的累加和

1. 在空白卡片 (x) 上写0
2. 取第一张卡片 (1)
3. 计算两张卡片上的数值的和
4. 擦除卡片x上的数值，并写入前一步的计算结果
5. 丢弃第一张卡片
6. 取当前的第一张卡片 (2)
7. ....
8. 报告卡片x上的值

哪些是数据？ 哪些是处理/计算？



# 数据和表示数据

- 数据：1..10、和（结果）
- 数据的使用方式
  - 在计算过程中观察：变化还是不变化
  - 1..10，仅提取卡片上的值
  - 和，卡片x，有擦除和重写的操作（变化）
- 表示数据
  - 1..10，直接使用常数
  - 和，变量，估计取值范围（0..100），选择数据类型int，命名x



# 重写计算过程

1. 在空白卡片 (x) 上写0
2. 取第一张卡片 (1)
3. 计算两张卡片上的数值的和
4. 擦除卡片x上的数值，并写入前一步的计算结果
5. 丢弃第一张卡片
6. 取当前的第一张卡片 (2)
7. ....
8. 报告卡片x上的值

1.  $x = 0;$

2.  $x = x + 1;$

3.  $x = x + 2;$

.....

12. 输出x中的数据; ?



# 编写源程序

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

```
int main()  
{
```

```
    int x;           /* 定义变量 */
```

```
    x = x + 1;        /* 逐步计算 */
```

```
    x = x + 2;
```

```
    x = x + 3;
```

```
    x = x + 4;
```

```
    x = x + 5;
```

```
    x = x + 6;
```

```
    x = x + 7;
```

```
    x = x + 8;
```

```
    x = x + 9;
```

```
    x = x + 10;
```

```
    printf("%d\n", x); /* 输出结果 */
```

```
    return 0;         /* 程序结束 */
```

```
}
```



# 数据展示

- ❖ 55
- ❖  $55=1+2+3+4+5+6+7+8+9+10$
- ❖  $1+2+3+4+5+6+7+8+9+10=55$
- ❖  $1+2+3+4+5+6+7+8+9+10 = 55$
- ❖  $1+2+\dots+10 = 55$
- ❖  $1+2+\dots+9+10 = 55$
- ❖  $1+\dots+10 = 55$
- ❖ 1到10的累加和是55



# 向用户展示数据（屏幕输出）

- 用户

- 不同的用户，有不同的观看数据的角度
- 相同领域的用户可能有近似的阅读方式和习惯

- 让用户看到数据

- 将数据在屏幕上显示出来

- 让用户看得明白

- 显示数据的同时，附加数据的相关说明

- 让用户看得习惯

- 让显示的数据符合用户的阅读方式和习惯



# printf: 格式化打印函数

- ❖ 将数据以设定的格式打印到屏幕
  - ❖ `void printf(格式控制字符串, 表达式[, [表达式[, ……]])`
- ❖ 函数的要素是?
  - ❖ 函数名: `printf`
  - ❖ 形式化参数列表: `格式控制字符串, 表达式[, [表达式[, ……]]]`
  - ❖ 返回值: 无



# 格式控制字符串（参数1）

- ❖ 字符串常量
- ❖ 表达式的格式控制字符串：`%[flags][width][.perc][F|N|h|l]type`
  - ❖ `[]`括起来的部分表示可忽略、`|`表示“或者”
- ❖ Flags “-+0”“-+ ”
  - ❖ 左对齐（-，左对齐将忽略前导符号）
  - ❖ 显示数符（+|-）
  - ❖ 前导符号（0|空格，填充左边的空白区域）
- ❖ `width.perc`
  - ❖ 输出宽度（屏幕上的字符位数，包含小数点和小数位数）和精度（小数位数）



# 格式控制字符串（参数1）

## ✧ F|N|h|l

✧ F: 远指针

N: 近指针

✧ h: 短整数或单精度浮点数

l: 长整数或双精度浮点数

## ✧ type

d 有符号10进制整数

E/e 用科学表示格式的浮点数

i 有符号10进制整数

G/g 使用%f和%e表示的总位数最短的来表示浮点数，G是以该数的指数形式表示

o 无符号8进制整数

c 单字符

u 无符号10进制整数

s 字符串

x 无符号的16进制数字，并以小写字母表示

S wchar\_t字符类型字符串

X 无符号的16进制数字，并以大写字母表示

% 显示百分号本身

F/f 浮点数

p 显示一个指针



# 改变示例的输出格式

- ❖ 55
- ❖  $55=1+2+3+4+5+6+7+8+9+10$
- ❖  $1+2+3+4+5+6+7+8+9+10=55$
- ❖  $1+2+3+4+5+6+7+8+9+10 = 55$
- ❖  $1+2+\dots+10 = 55$
- ❖  $1+2+\dots+9+10 = 55$
- ❖  $1+\dots+10 = 55$
- ❖ 1到10的累加和是55



# 问题：求任意两个整数的和

- $z = x + y;$
- 数据
- 过程
  - 开始;
  - 获取第一个数据(?);
  - 获取第二个数据(?);
  - 计算两的数据的和;
  - 得到结果;
  - 结束。



# 从键盘输入数据

- ❖ scanf函数

- ❖ 调用规约

- ❖ scanf(“格式控制字符串”, 输入项地址表);

- ❖ 输入项地址表是一个或多个变量的地址

- ❖ 格式控制字符串

- ❖ %[宽度] [F|N] [h|l] 类型字符



# 输入格式控制

- 宽度(n)

- 读取键盘输入数据中相应的n位，但按需要的位数赋给相应的变量，多余部分被舍弃。

- 格式修饰符

- m 表示数据占用的宽度
  - l 加在d、o、x、u前：输入长整型
  - 加在f、e前：输入双精度型
  - L 加在f、e前：输入long double型
  - h 加在d、o、x前：输入短整型



# 输入格式控制

- 类型字符

- 表示输入后转换的数据类型。

- 与printf函数格式中的格式指示符相同。

- d 以带符号的十进制形式读入一个整数

- o 以八进制无符号形式读入一个整数

- x(X) 以十六进制无符号形式读入一个整数

- u 以无符号十进制形式读入一个整数

- c 以字符形式读入一个字符

- s 读入一个字符串

- f 以小数形式读入一个单精度数

- e(E) 以标准指数形式读入一个单精度数



# 输入格式控制

- 其他字符

- 空白字符

- 作为相邻2个输入数据的缺省分隔符；

- 非空白字符（普通字符）

- 普通字符不是显示的而是规定在输入有效数据时，必须原样一起输入的字符。



# 示例

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

```
int main()
```

```
{
```

```
    int x;        /* 定义变量 */
```

```
    int y;
```

```
    int z;
```

```
    scanf("%d", &x);        /* 输入数据 */
```

```
    scanf("%d", &y);
```

```
    /* scanf("%d%d", &x, &y); */
```

```
    z = x + y;    /* 计算 */
```

```
    printf("%d + %d = %d\n", x, y, z); /* 输出结果 */
```

```
    return 0;        /* 程序结束 */
```

```
}
```



# 注意事项

- scanf () 的格式控制字符串中的普通字符不是用于输出的,而是要求按原样进行输入。
  - scanf ("x=%d", &x) ;
- 参数的第二部分一定是地址列表,不能是表达式。
- 执行scanf()输入数据时,在两个数据之间允许以一个或多个空格间隔,也可以用回车键、tab键分隔。
- 实数不许规定精度,像%10.4f是不合法的。
- 如果输入时类型不匹配则停止处理,返回0。



# 其它输入/输出函数

- ❖ 字符与字符串输入函数

- ❖ getch/getchar/gets

- ❖ getche

- ❖ 字符与字符串输出函数

- ❖ putch/putchar/puts



# 单个字符的输入与输出

- ❖ 字符常量

- ❖ 用单引号括起来的一个字符

- ❖ 'd' 'D' '0'

- ❖ 转义字符 \

- ❖ '\n' '\r' '\0' '\t' '\v' '\b' '\a' '\"' '\\' '\?' '\ddd' '\xhh'

- ❖ ddd是三位八进制数，hh是2位十六进制数，表示某个字符的ASCII码值



# 字符输入与输出

- ❖ 从键盘输入一个大写英文字母，将其转换为小写字母，再显示到屏幕上
- ❖ 输入 `getchar scanf`
- ❖ 输出 `putchar printf`
- ❖ 如何转换
  - ❖ ASCII
  - ❖ 字符是一个数值 (8bit、1Byte)



# scanf使用时的问题

■ P55