



# C 语言和程序设计思想 I



# 内容

- 字符串的格式化输入与输出
- 运算符、表达式和语句
- C 控制语句: 循环



# 数组

- 数组： **同一类型**数据元素的**有序**序列
- 存储形式：连续
- 应用方式：下标引用

e.g. `int Num[5] = {1, 2, 3, 4, 5};`

`Num[1] ?   Num[4] ?   Num[5] ?   1 ?`



# 字符串 (string)

- 类型: char 数组
- e.g.

```
char[ ] = "Interesting";
```

- 空字符: \0
- 格式说明符: %s
- ‘a’ 和 “a” 区别



# 字符串的输出

- 函数: `printf( );`
- 输出整个字符串
- 输出字符串中的指定字符
- 输出长字符串



# 字符串的输入

- 函数: scanf();
- 格式化输入字符串



# printf 和 scanf

- 返回值

e.g.

```
int x = scanf("%d %d", &a, &b); --输入的变量数  
int y = printf("%d\n%d\n", a, b); --输出的字符数
```



# string.h

- strlen()
- 作用： 获得字符串长度
- 使用
- 对比： strlen 和 sizeof





# 分析程序

```
#include <stdio.h>
#include <string.h>
#define DENSITY 62.4
int main (void)
{
    float weight, volume;
    int size, letters;
    char name[40];    // name 是一个有 40 个字符的数组

    printf("Hi! What's your first name?\n");
    scanf("%s", name);
    printf("%s, what's your weight in pounds?\n", name);
    scanf("%f", &weight);
    size = sizeof name;
    letters = strlen(name);
    volume = weight / DENSITY;
    printf("Well, %s, your volume is %2.2f cubic feet.\n", name, volume);
    printf("Also, your first name has %d letters, \n", letters);
    printf("and we have %d bytes to store it in.\n", size);
    return 0;
}
```



# 常量

- const
- #define



# 内容

- 字符串的格式化输入与输出
- 运算符、表达式和语句
- C 控制语句: 循环



# 基本运算符

- =
- + -
- \* / %
- ++ --
- sizeof
- ( )



# 复合运算符

- $+=$
- $-=$
- $*=$
- $/=$
- $\%o=$



# 关系运算符

- $>$

- $<$

- $>=$

- $<=$

- $==$

- $!=$



# \*指派运算符

- (type)

e.g. `mice = 1.6 + 1.7`

`mice = (int)1.6 + (int)1.7`

- typedef

e.g. `typedef int ElemType`



# \*左值和右值

- 左值—运算符左边—可修改
- 右值—运算符右边—能赋给左值





# 内容

- 字符串的格式化输入与输出
- 运算符、表达式和语句
- C 控制语句: 循环



# 三种循环结构

- for
- while
- do...while



# while 循环

- while(expression)  
statement
- expression -> 判断是否执行循环体的表达式
- statement -> 程序语句



# for 循环

- for(initialize; test; update)  
statement
- initialize -> 计数变量初始化
- test -> 判断是否执行循环体的表达式
- update -> 更新计数变量的值



# \*逗号运算符

- 多个初始化或者多个更新表达式

e.g. `for(i = 0, j = 10; i < j; i++, j--)`



# do...while 循环

- do  
    statement  
while(expression);
- 至少循环一次



# 三种循环结构比较

- while — 入口条件循环
- for — 计数循环
- do...while — 退出条件循环



# 循环体结构的选择

- 入口? 退出?
- 至少必须循环一次?
- while 和 for?





# 真假判断

- 真值：非零值
- 0 为假



# 运算符的优先级

- 赋值运算符 < 关系运算符 < 算术运算符

- e.g.  $x > x + 2$

$$y < x - 3$$
$$(x > 3) + 1$$
$$x = 2 < -1$$



# scanf 在循环体中的使用

```
int x = 0;
while(x < 10)
{
    scanf("%d",&x);
    printf("%d\n", x);
}
```

```
int x;
while(scanf("%d",&x))
{
    printf("%d\n", x);
}
```



# 练习

- 使用循环输出数组 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
- 使用循环输出字符串“苟利国家生死以”
- 计算 5!
- 计算 100 以内偶数的和



# 作业

- 第四章 4.7-2、4、6、9、11；4.8-4、7
- 第五章 5.10-1、3、5、9；5.11-1、8
- 第六章 6.15-5、6、10；6.16-4、14
- 重点：字符串的输入输出、运算符、三种循环结构