



清华大学

腾讯会议 ID: 948 606 031

Tsinghua University

计算机程序设计基础

第3讲 顺序程序设计与 基本数据类型（二）

沈瑜 (010-62782951)

shenyu@tsinghua.edu.cn

清华大学电机系

2022.9.27



设整型变量 $a=2$ ，则执行下列语句后，浮点型变量 b 的值为 0.5 的是：

- ☒ A $b=1.0/a;$
- ☐ B $b=(float)(1/a);$
- ☒ C $b=1/(float)a;$
- ☒ D $b=1/(a*1.0);$

提交





主要内容

- 简单的**C**程序设计举例（二）
 - 顺序程序设计，实现简单的输入输出
- 简单的数据类型（二） 字符型数据
- 简单数据的输入输出（二）
 - **printf**
 - **scanf**
 - **putchar**
 - **getchar**

《C程序设计（第五版）》第3章

《C语言程序设计：现代方法(第2版)》第2,3,4,7章





3.1 简单的C程序设计举例（二）

● 例1: TalkMachine Ver1.0

```
#include <stdio.h>

#define DENSITY_AVERAGE  1.0459 //人的密度，单位是：公斤/每立方分米

int main()
{
    float weight, volume;
    char name[40];

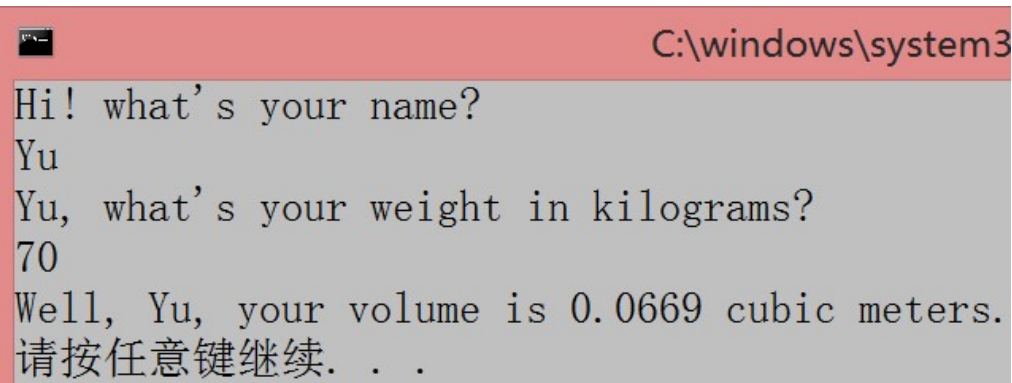
    printf("Hi! what's your name?\n");
    scanf( "%s", name );

    printf("%s, what's your weight in kilograms?\n", name);
    scanf("%f", &weight);

    volume = weight / DENSITY_AVERAGE /1000.;

    printf("Well, %s, your volume is %.4f cubic meters. \n", name, volume);

    return 0;
}
```



```
C:\windows\system32
Hi! what's your name?
Yu
Yu, what's your weight in kilograms?
70
Well, Yu, your volume is 0.0669 cubic meters.
请按任意键继续. . .
```





●例2: 成绩的等级

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

```
int main()
```

```
{
```

```
    int score;
```

```
    printf("你好! 你的《计算机程序设计基础》成绩预计得多少分? \n");
```

```
    scanf( "%d", &score );    //输入成绩
```

```
    char grade; //以下7/8行, 是通过选择语句, 将成绩分数转化为等级。学习选择语句时再详细研究。
```

```
    if( score >= 93 && score <=100 )
```

```
        grade = 'A';
```

```
    else if( score >= 85 && score <=92 )
```

```
        grade = 'B';
```

```
    else if( score >= 77 && score <=84 )
```

```
        grade = 'C';
```

```
    else if( score >= 70 && score <=76 )
```

```
        grade = 'D';
```

```
    else if( score <= 69 )
```

```
        grade = 'F';
```

```
    printf("你的预计成绩等级为: %c \n", grade );    //输出等级
```

```
    return 0;
```

```
}
```

```
C:\windows\system32\cmd
你好! 你的《计算机程序设计基础》成绩预计得多少分?
99
你的预计成绩等级为: A
请按任意键继续. . .
```





不识字的计算机？

- 例3：问题：字符，是怎么存储/使用的？

```
printf("Hello!\n");  
printf("你好！ \n");
```





3.2 简单的数据类型（二）字符型数据

1. 字符与字符代码 (American Standard Code for Information Interchange)

- 大多数系统采用标准ASCII字符集（共**128**个）
 - 字母：A ~ Z a ~ z
 - 数字：0 ~ 9
 - 专门符号：29个：! " # & ' () * 等
 - 空格符：空格、水平/垂直制表符、换行、换页
 - 不能显示的字符：空(null)字符(以'\0'表示)、警告(以'\a'表示)、回车(以'\r'表示)、换行(以'\n'表示)等
- 每个字符在计算机内部用**一个整数**表示，称为**字符编码**
 - 参见P.377 的ASCII码表（标准/扩展）



ASCII value	Character	Control character	ASCII value	Character	ASCII value	Character	ASCII value	Character
000	(null)	NUL	032	(space)	064	@	096	
001	☺	SOH	033	!	065	A	097	a
002	☻	STX	034	"	066	B	098	b
003	♥	ETX	035	#	067	C	099	c
004	♦	EOT	036	\$	068	D	100	d
005	♣	ENQ	037	%	069	E	101	e
006	♠	ACK	038	&	070	F	102	f
007	(beep)	BEL	039	'	071	G	103	g
008	■	BS	040	(072	H	104	h
009	(tab)	HT	041)	073	I	105	i
010	(line feed)	LF	042	*	074	J	106	j
011	(home)	VT	043	+	075	K	107	k
012	(form feed)	FF	044	,	076	L	108	l
013	(carriage return)	CR	045	-	077	M	109	m
014	♪	SO	046	.	078	N	110	n
015	☼	SI	047	/	079	O	111	o
016	▲	DLE	048	0	080	P	112	p
017	▼	DC1	049	1	081	Q	113	q
018	↕	DC2	050	2	082	R	114	r
019	!!	DC3	051	3	083	S	115	s
020	π	DC4	052	4	084	T	116	t
021	\$	NAK	053	5	085	U	117	u
022	▬	SYN	054	6	086	V	118	v
023	↕	ETB	055	7	087	W	119	w
024	↑	CAN	056	8	088	X	120	x
025	↓	EM	057	9	089	Y	121	y
026	→	SUB	058	:	090	Z	122	z
027	←	ESC	059	;	091	[123	{
028	(cursor right)	FS	060	<	092	\	124	
029	(cursor left)	GS	061	=	093]	125	}
030	(cursor up)	RS	062	>	094	^	126	~
031	(cursor down)	US	063	?	095	_	127	␣

字符的表示

- 用单引号括起来一个字符
 - 如: 'A' 'a' ' ' '?' '1' ''
- 有的字符是不可见的, 只能用特殊形式(转义字符)表示。下面几个一定要牢记:
 - 如: '\n' '\r' '\t' '\"' '\''' '\\'
 - 如: '\012' '\x0A' 八进制、16进制表示
- 字符'1'和整数1是不同的概念:
 - 字符'1'只是代表一个形状为'1'的符号, 需要时按原样输出, 在内存中以ASCII码形式存储, 占1字节 //值为49

0 0 1 1 0 0 0 1

- 整数1是以整数存储方式(二进制补码方式)存储的, 占2个或4个字节

0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1
-----------------	-----------------



2. 字符变量

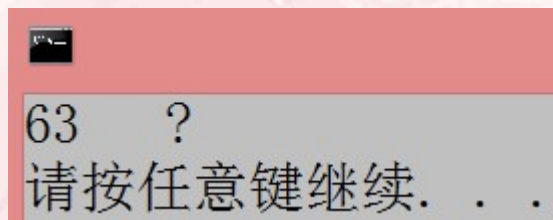
- 用类型符**char**定义字符变量

```
char c = '?';
```

定义字符变量**c**，且初始化为 '?'

```
printf("%d %c\n", c, c);
```

输出结果是：



```
63 ?  
请按任意键继续. . .
```

3. 字符串常量

- 一对双引号括起来的字符序列。“China”
 - 注意：C在每个字符串结尾处加上字符 ‘\0’
 - 因此，“a” 包括两个字符：‘a’ ‘\0’
 - “China”包括6个字符：
‘C’ ‘h’ ‘i’ ‘n’ ‘a’ ‘\0’



4. 字符变量的赋值和运算

- 字符的本质是整数（ **-128~127** 或 **0~255** ）
- 例1： 字符赋值和运算示例

```
char c = 'K'; //定义字符变量c，且初始化为 'K'
```

```
c++;          //自增运算
```

```
printf("%d %c\n", c, c);
```

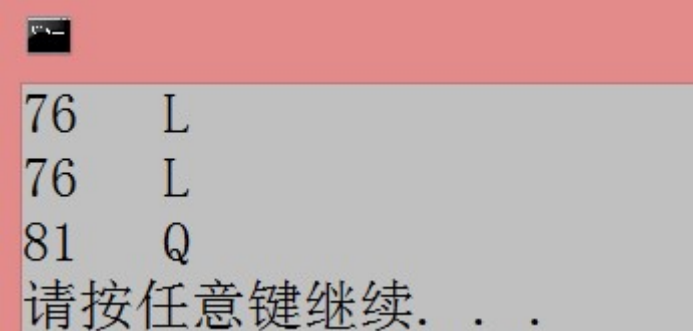
```
c = c + 256; //加法运算、赋值
```

```
printf("%d %c\n", c, c);
```

```
c = c + 5;
```

```
printf("%d %c\n", c, c);
```

输出结果是：



```
76 L
76 L
81 Q
请按任意键继续. . .
```



字符处理的若干技巧

- 例2 技巧1: 大小写字母转换

`char c = 'm' + 'A' - 'a';`

'm'转换为'M'

- 例3 技巧2: 数字字符转换为相应的整数

`int k = '9' - '0';`

'9'转换为 9

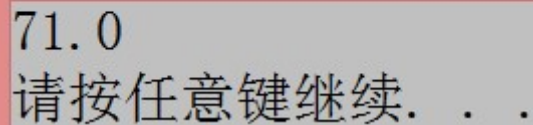


字符的运算

- 例4：计算“CHINA”中各字母的平均值

```
printf("%.1f\n", ('C'+'H'+'I'+'N'+'A')/5.);
```

输出结果是：



```
71.0  
请按任意键继续. . .
```

- 如果按以下方法呢？试试结果、并做解释！

```
printf("%.1f\n", ('C'+'H'+'I'+'N'+'A')/5);
```





3.3 简单数据的输入输出（二）

1. printf函数的用法(续)

printf (格式控制, 输出表列)

- c 格式符。用来输出一个字符

```
char ch='a';
```

```
printf("%c",ch); 或
```

```
printf("%5c",ch);
```

- s 格式符。用来输出一个字符串

```
printf( "%s", "CHINA" );
```

输出字符串: **CHINA**

**p74 表3.6, 3.7其他
格式符, 请自学**



2. scanf函数的用法

scanf(格式控制, 地址表列)

含义同**printf**函数

可以是变量的地址,
或字符串的首地址

scanf函数中的格式声明

- 与**printf**函数中的格式声明相似
- 以%开始, 以一个格式字符结束, 中间可以插入附加的字符

```
scanf("a=%f,b=%f,c=%f",&a,&b,&c);
```



使用scanf函数时应注意的问题

◆在回车**ENTER**键之后，才正式输入；此前缓存

◆格式控制后，应给出变量地址，而不是变量名

`scanf("%f%f%f",a,b,c);` //错

`scanf("%f%f%f",&a,&b,&c);` //对

◆**%d**或**%f**格式控制输入时，需用空格或逗号分隔
(默认**1**个或更多地空格)

`scanf("%f%f%f",&f1, &f2, &f3);`

1 3 2↵ 对

1,3,2↵ 错



scanf

◆ 格式控制中的普通字符，需原样输入：

`scanf("a=%f,b=%f,c=%f",&a,&b,&c);`

1 3 2↵

错

a=1,b=3,c=2↵

对

a=1 b=3 c=2↵

错

◆ %c 格式控制时，不要输入空格

对于 `scanf("%c%c%c",&c1,&c2,&c3);`

abc↵

对

a b c↵

错

对于 `scanf("%f%c%f",&a,&b,&c);`

若输入

1234+2345.26

注意：编译程序不检查 表列数目

◆ printf和scanf不检查输出表列的数目

```
int i=3, j=5;
```

```
printf( "%d %d", i ); /* WRONG 无意义的输出 */
```

```
printf( "%d", i, j, i+1, j+1 ); /* WRONG 少输出 */
```

◆ printf和scanf不检查输出变量类型与占位符是否匹配

```
int i=3; float j=5.0;
```

```
printf( "%f %d", i, j ); /* WRONG 无意义的输出 */
```

◆ printf和scanf的重要差别

- printf 往往以'\n'结尾，scanf不要有'\n' //输入混乱
- printf用输出变量名，scanf要用变量地址 &



3. 用**putchar**函数输出一个字符

- 从计算机向显示器输出一个字符
- **putchar**函数的一般形式为：

putchar(c)



例： 先后输出**BOY**三个字符

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

```
int main ( )
```

```
{
```

```
    char a='B',b='O',c='Y';
```

```
    putchar(a);      //向显示器输出字符'B'
```

```
    putchar(b);
```

```
    putchar(c);
```

```
    putchar ('\n');  //向显示器输出换行符
```

```
    return 0;
```

```
}
```

BOY



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

```
int main ( )
```

```
{
```

```
    int a=66,b=79,c=89;//char a='B',b='O',c='Y';
```

```
    putchar(a);
```

//向显示器输出字符'B'

```
    putchar(b);
```

```
    putchar(c);
```

```
    putchar ('\n');
```

//向显示器输出换行符

```
    return 0;
```

```
}
```

改为int a=66,b=79,c=89;

BOY



4. 用**getchar**函数输入一个字符

用**getchar**函数输入一个字符：

- 向计算机输入一个字符
- **getchar**函数的一般形式为：

getchar()



- 例3.9 从键盘输入**BOY**三个字符，然后把它们输出到屏幕。

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

```
int main ( )
```

```
{ char a,b,c;
```

```
    a=getchar();           //输入一个字符，送给变量a
```

```
    b=getchar();
```

```
    c=getchar();
```

```
    putchar(a); putchar(b); putchar(c);
```

```
    putchar('\n');
```

```
    return 0;
```

```
}
```

BO
BO



- 例**3.9** 从键盘输入**BOY**三个字符，然后把它们输出到屏幕。

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

```
int main ( )
```

```
{
```

```
    putchar(getchar());
```

```
    putchar(getchar());
```

```
    putchar(getchar());
```

```
    putchar('\n');
```

```
    return 0;
```

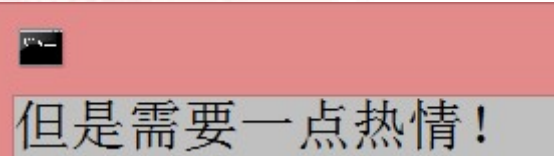
```
}
```



BOY
BOY



```
#include <stdio.h>
int main()
{
    printf("学习C语言很容易, \r");
    printf("但是需要一点热情! \n");
    return 0;
}
```



但是需要一点热情!

为什么会少掉一行?



```
int a=1234;  
float b=123.456; double c=12345.54321;  
printf("%2d, %2.1f, %2.1f\n",a,b,c);
```

执行后的结果为：

- ☐ A 输出格式中位数不够
- ☐ B 输出结果为12, 12.4, 12.5
- ☒ C 输出结果为1234, 123.5, 12345.5
- ☐ D 输出结果为1234, 123.4, 1234.5

提交

