



## [2] 顺序结构程序设计

深入浅出程序设计竞赛  
第 1 部分 – 语言入门  
V 2021-04

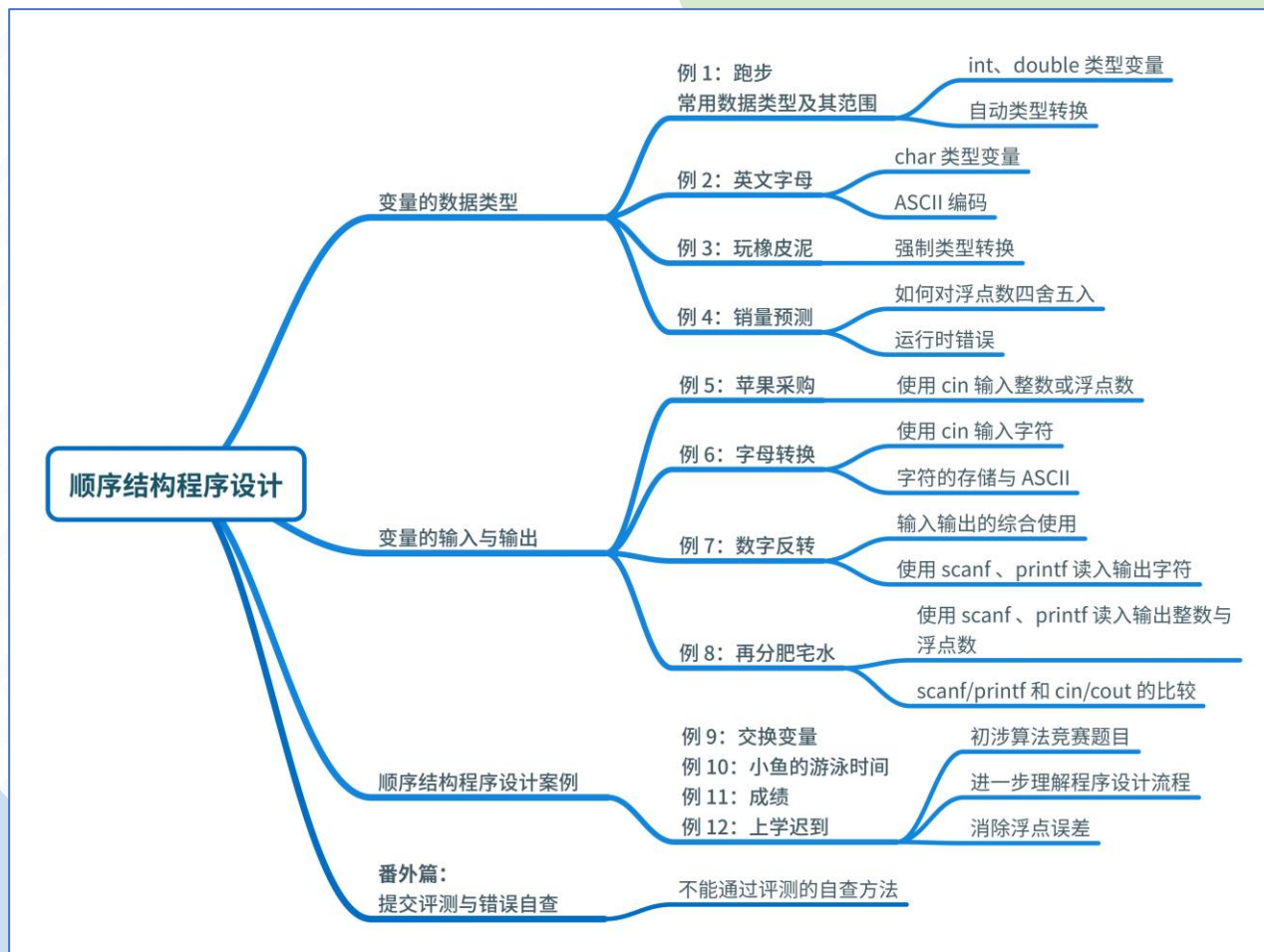
## 版权声明

本课件为《深入浅出程序设计竞赛 - 基础篇》的配套课件，版权归 洛谷 所有。所有个人或者机构均可免费使用本课件，亦可免费传播，但不可付费交易本系列课件。

若引用本课件的内容，或者进行二次创作，请标明本课件的出处。

- 其它《深基》配套资源、购买本书等请参阅：  
<https://www.luogu.com.cn/blog/kkksc03/IPC-resources>
- 如果课件有任何错误，请在这里反馈  
<https://www.luogu.com.cn/discuss/show/296741>

# 本章知识导图



## 第 2 章 顺序结构程序设计

---

变量的数据类型

变量的输入与输出

顺序结构程序设计案例

提交评测与错误自查

课后习题与实验

# 变量的数据类型

C++有很多种变量类型，能够储存的数据精度和范围都不一样，所以大家要根据数据的实际情况选择合适的变量类型。

请翻至课本 P15

# 数据类型

有时将要处理的**数据**临时储存起来备用，存在对应的**变量**。不用种类的数据（比如整数、字符、浮点数等）都有对应的**数据类型**，精度和范围不一样。这些变量放在计算机的**内存**中。

以做菜为例子：

- 数据：原材料食材，随时需要用
- 变量：装食材的容器
- 数据类型：适用于不同食材的容器（盘子、水桶、菜筐）
- 内存：操作台

# 跑步

## 例 2.1

小A跑步速度  $5\text{m/s}$ ，八尾勇跑步速度  $8\text{m/s}$

八尾勇在小A后面  $100\text{m}$ ，他们同时起跑，请问需要多长时间八尾勇可以追上小A？

输出一个数字表示答案。

提示：八尾勇和小A之间的相对速度是  $(8-5)=3\text{m/s}$ ，也就是说每秒钟他们的相对距离就会缩短  $3\text{m}$ 。因此  $100/3$  就是答案

# 跑步

int 已经不能完全解决问题，于是就要使用一种新的数据类型：

double 类型的意思是双精度浮点数，暂时可以认为是能存放带小数的数字，8字节，有效数字15位。

```
#include<iostream>
using namespace std;
int main() {
    int v_a = 5, v_yao = 8, distance = 100;
    // 小A和八尾勇的速度，以及距离
    double delta, ans; // 速度的差值和答案
    delta = v_yao - v_a;
    // 两人相对速度，也就是每秒距离缩短多远
    ans = distance / delta;
    // ans = 1.0 * distance / (v_yao - v_a)
    cout << ans << endl;
    return 0;
}
```



# 跑步

定义 3 个 `int` 类型的变量，并赋值初始值

```
int v_a = 5, v_yao = 8, distance = 100;
```

定义 2 个 `double` 类型的变量，不赋值初始值

```
double delta, ans;
```

`int` 和 `double` 混合运算，`int` 会被 类型转换 为 `double`

如果要手动将 `int` 类型的 `a` 变量变为 `double` 类型进行计算，

可以，`a*1.0` 或者 `(double)a`，后者是强制类型转换

其他常见的数据类型和数据范围 见课本P16。

```
ans = distance / delta;  
// ans = 1.0 * distance / (v_yao - v_a)
```

# 常见的数据类型

数据类型	占用空间	取值范围
char	1 字节, 8 位	$-128 \sim 127$
int	4 字节, 32 位	$-2^{31} \sim 2^{31} - 1$ , 大约能够表示绝对值不超过 $2.1 \times 10^9$ 的整数
unsigned int	4 字节, 32 位	$0 \sim 2^{32} - 1$ , 大约能够表示不超过 $4.2 \times 10^9$ 的非负整数
long long	8 字节, 64 位	$-2^{63} \sim 2^{63} - 1$ , 大约能够表示绝对值不超过 $9.2 \times 10^{18}$ 的整数
unsigned long long	8 字节, 64 位	$0 \sim 2^{64} - 1$ , 大约能够表示不超过 $1.8 \times 10^{19}$ 的非负整数
float	4 字节, 32 位	大约指数绝对值不超过 37, 6 位有效数字
double	8 字节, 64 位	大约指数绝对值不超过 307, 15 位有效数字

# 英文字母

## 例 2.2

大家都知道有 26 个英文字母，其中 A 是第一个字母。现在请编程求出：

- 1、M 是字母表中的第几个字母？
- 2、第 18 个字母是什么？

输出两个答案，使用换行隔开。

**提示：**这时我们发现，题目中涉及了字母，而我们刚才学习的数据类型都是存储整数的，所以现在需要一种新的存储类型。

# char

---

char 是一种存储字符的变量。

char 类型的本质就是一个不大于 127 的整数，只是这个整数可以表现成一个对应的字符。具体的表现方法已经进行了规定，具体的对应关系可见 课本P17。

## ASCII码

数字	字符	数字	字符	数字	字符	数字	字符	数字	对应字符	数字	字符
32	[空格]	48	0	64	@	80	P	96	`	112	p
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	\$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	(	56	8	72	H	88	X	104	h	120	x
41	)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[	107	k	123	{
44	,	60	<	76	L	92	\	108	l	124	
45	-	61	=	77	M	93	]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o		

# 英文字母

大写字母、小写字母和数字在 ASCII 表中都是按照顺序依次排列的。

```
ans1 = 'M' - 'A' + 1;
```

'M'-'A'+1 是计算字母 'M' 和字母 'A' 中间的差距。

ans1 等价于  $77 - 65 + 1$ 。77 和 'M' 等价，65 和 'A' 等价。

```
ans2 = 'A' + 18 - 1;
```

'A'+18-1：计算字母 'A' 后面的第 18-1 个字母。

ans2 是  $65+18-1$  也就是 82。

注意单个 char 字符外面用单引号包裹

# 英文字母

```
#include<iostream>
using namespace std;
int main() {
    int ans1;
    char ans2;
    ans1 = 'M' - 'A' + 1;
    ans2 = 'A' + 18 - 1;
    cout << ans1 << endl;
    cout << ans2 << endl;
    return 0;
}
```

使用 cout 输出 char 类型的变量，会输出这个变量存储的数字对应的字符。

如何输出 82 这个数字？

1. 把这个变量赋值给 int 类型
2. 使用接下来提到的类型强制转换方法

# 玩橡皮泥

## 例 2.3

小 A 有两块球形橡皮泥，一个半径是 4，一个半径是 10。他想把这两块橡皮泥揉在一起，然后塑造成一个正方体，请问这个正方体的边长是多少？

若结果非整数，则舍去小数点后的数字。取  $\pi = 3.141593$ 。

已知球半径求体积的公式是  $V = \frac{4}{3}\pi r^3$ 。所以总体积是  $\frac{4}{3}\pi(4^3 + 10^3)$ ，其正方体的边长就是就是体积立方根，所以最后结果是

$$\sqrt[3]{\frac{4}{3}\pi(4^3 + 10^3)}。$$

如果数学课上没有学过相关知识看不懂，跳过这题也是可以的！



# 玩橡皮泥

最后结果是  $\sqrt[3]{\frac{4}{3}\pi(4^3 + 10^3)}$ 。要使用 `pow()` 函数计算立方根。

为保证结果准确，在最后一步输出之前，要使用浮点数！

整数变浮点？

- 后面加 `.0` 或 `*1.0`
- 强制类型转换 `(double)r1`

浮点变整数？

- 直接赋值给整数变量
- 强制类型转换 `(int)l`

重申：数学课上没有学过相关知识看不懂，跳过这题也是可以的！

```
#include<iostream>
#include<cmath>
using namespace std;
#define PI 3.141593
int main() {
    int r1 = 4, r2 = 10;
    double V, l;
    V = 4.0/3 * PI *(r1*r1*r1 + r2*r2*r2);
    // 计算体积
    l = pow(V, 1.0 / 3);
    // 使用立方根计算边长，注意不能写成1/3
    cout << (int)l << endl;
    return 0;
}
```

正确使用数据类型是第一个难点

# 变量的输入与输出

在运行程序的时候告诉程序这需要处理的数字，程序可以返回答案。为了达成这个目的，必须让程序接受输入，并让程序能够处理这些输入。

请翻至课本 P20

## 输入 cin/cout

到现在为止的程序都是没有输入的。这相当于走进厨房执行“做菜”时，发现原材料已经放在碗里准备好了。

于是我们就要使用 cin 来告诉计算机一些数据。输入多个数据，使用换行或者空格（可以不止一个）隔开。

一个例子：

```
cin >> t >> n;
```

这里的 t 和 n 就是我们之前学习过的变量。

# 苹果采购

## 例 2.5

现在需要采购一些苹果，每名同学都可以分到固定数量的苹果，并且已经知道了同学的数量，请问需要采购多少个苹果？

从键盘输入数据。

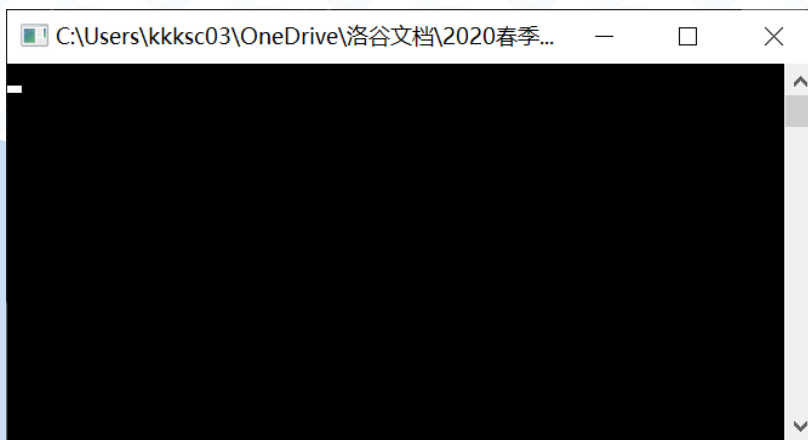
注意：在这里将用户输入给程序的形式可能有所差异

样例输入：3 4

样例输出：12

# 苹果采购

```
#include<iostream>
using namespace std;
int main() {
    int t, n;
    //每个人的苹果数量 和 学生数量
    cin >> t >> n;
    cout << t * n << endl;
    return 0;
}
```



我们可以写出如下程序。

运行程序……发现除了一个黑框框什么都没有。

不用慌，你还没告诉计算机有每人几个苹果和几名同学呢！

注意！是本地运行，不是在线 IDE！

在线 IDE 的输入 3 4，并且按一下运行。

# 苹果采购

于是在黑框框中输入 3 4，并且按一下回车键。

意思是每人 3 个苹果，有 4 名同学。

计算机马上给出了结果：12

每位一共需要采购 12 个苹果，答案正确。

```
int t, n;  
cin >> t >> n;  
cout << t * n << endl;
```

首先定义两个 `int` 类型的变量 `t` 和 `n`，分别表示每人可以分到的苹果数量和 student 数量。最后输出了 `t*n` 的值表示答案。

# 数字反转

---

## 例 2.7 (洛谷P5705)

输入一个不小于 100 且小于 1000，同时包括小数点后一位的一个浮点数。

例如 123.4，要求把这个数字翻转过来，变成 4.321 并输出。

# 数字反转

**解法 1:** 分离出这个数字的所有位数，乘 10 把这个它变成一个四位数整数，模 10 取余数，获得这个四位数的个位数字

通过类似的办法获得这个四位数的其他位数

```
#include<iostream>
using namespace std;
int main() {
    double p; // 输入的数字
    int q, a, b, c, d; // 转换成的四位数和分离出来的4位数字
    cin >> p;
    q = int(p * 10);
    a = q / 1000; // 千位
    b = q / 100 % 10; // 百位
    c = q / 10 % 10; // 十位
    d = q % 10; // 个位
    cout << d << "." << c << b << a << endl;
    return 0;
}
```



# 数字反转

解法 2：既然学习过了 char 类型，那么可以把输入视为 5 个字符，然后直接输出。

```
#include<iostream>
using namespace std;
int main() {
    char a, b, c, dot, d;
    cin >> a >> b >> c >> dot >> d;
    cout << d << dot << c << b << a << endl;
    return 0;
}
```

# scanf/printf 占位符

---

用于 格式化输入输出 数据的方法之一。

可以指定不同格式，如保留多少位浮点数、输出的数据占多宽的空间，等等。

这些占位符常用于 C 语言风格的输入输出方式，即 scanf/printf 。

这些占位符，都以百分号“%”为开头。

# 常见输入输出占位符

占位符	说明
%d	一个十进制整数，一般用于int类型（最常用）
%nd(n是正整数)	输出一个整数，如果不足n位，前面用空格补齐直到够n位。
%l64d(Windows), %lld(Linux)	一个十进制整数，一般用于long long类型； 要非常注意在不同的系统下，这个占位符是不一样的
%f	读入一个float类型的带小数点的浮点数。 或者输出float或者double类型的浮点数，默认 6 位小数。
%lf	读入double类型的浮点数。 注意输出double类型不用这个而是 %f。
%.nf(n是正整数)	用于输出一个固定n位小数的浮点数
%0nd(n是正整数)	输出一个整数，如果不足n位，前面用0补齐直到够n位。
%c	一个char类型的字符
%s	一个字符串

## 其他格式化方式

---

对于 `cout` 类型的输出，可以用 `fixed()` 和 `setprecision()` 来保留指定位数的浮点数。

`scanf/printf` 快于 `cin/cout`，尤其是数据量较大（百万级别）时速度差距会相当明显。

根据洛谷提交代码统计，使用 C 风格的 `scanf/printf` 数量要多于 C++ 的 `cin/cout`。

# 数字反转

## 例 2.7 (洛谷 P5705)

输入一个不小于 100 且小于 1000，同时包括小数点后一位的一个浮点数，例如123.4，要求把这个数字翻转过来，变成 4.321 并输出。要求使用 scanf/printf 完成此题。

解法3：使用 %c 读入依次字符，再倒序输出即可。

```
#include<cstdio>
using namespace std;
// 因为没有使用到C++语言的特性，这句话不加也没什么问题
int main() {
    char a, b, c, d;
    scanf("%c%c%c%c", &a, &b, &c, &d);
    printf("%c.%c%c%c", d, c, b, a);
    return 0;
}
```

## 再分肥宅水

### 例 2.8 (洛谷 P5706)

有  $t$  毫升水均分给  $n$  名同学。每名同学需要 2 个杯子。

问每名同学可以获得多少毫升饮料（严格精确到小数点后 3 位），以及一共需要多少个杯子。

输入一个实数  $t$  和一个整数  $n$ 。输出两个数字表示答案。

解析：使用浮点数进行运算，输出时指定了精度。

```
#include<cstdio>
using namespace std;
int main() {
    double t; int n;
    scanf("%lf%d", &t, &n);
    printf("%.3f\n%d", t / n, n * 2);
    return 0;
}
```

可以使用占位符 `%lf` 输入，`%.3f` 输出。

## 再分肥宅水

这道题需要使用浮点数进行运算，输出时指定了精度。  
可以使用占位符 `%lf` 输入，`%.3f` 输出。

```
#include<cstdio>
using namespace std;
int main() {
    double t;
    int n;
    scanf("%lf%d", &t, &n);
    printf("%.3f\n%d", t / n, n * 2);
    return 0;
}
```

# 顺序结构程序设计案例

其中的一些题目是各类比赛的真题（但是现在遇到的都很简单，不用担心！）

请翻至课本 P24



# 交换变量

## 例 2.9

定义两个变量  $a$  和  $b$  并输入两个数字存储进它们。

交换这两个变量并输出。

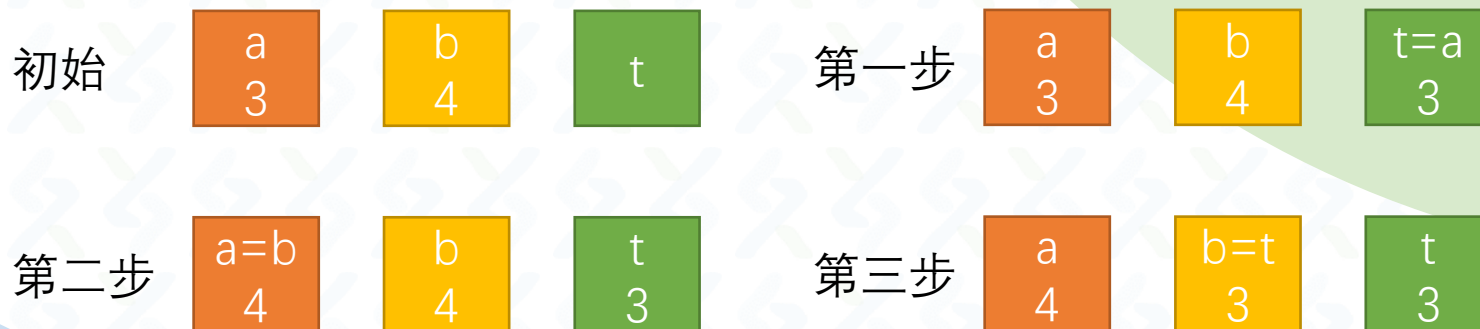
$a=b; b=a;$  行不行呢?

假设给你一杯可乐，一杯雪碧，要如何把这两个杯子中的饮料给交换呢?

# 交换变量

你需要准备第三个碗

- 将第一个碗的东西放进第三个碗
- 然后将第二个碗的东西放入第一个碗
- 最后把第三个碗的东西放回第二个碗



```
int a, b, t;  
scanf("%d %d", &a, &b);  
t = a; a = b; b = t; // 一组相关的短语句也可以写在一行内  
printf("%d %d", a, b);
```

# 算法竞赛的基本格式

一般来讲，对于输入的内容，每个数字之间以空格或者回车隔开。  
如图，是洛谷题目 P1425 的输入样例。

请仔细阅读输入格式和输出格式，了解如何输入输出。

## 输入格式

一行内输入 4 个整数，分别表示  $a, b, c, d$ 。

## 输出格式

一行内输出 2 个整数  $e$  和  $f$ ，用空格间隔，依次表示小鱼这天一共游了多少小时多少分钟。其中表示分钟的整数  $f$  应该小于 60。

## 输入输出样例

输入 #1

复制

12 50 19 10

输出 #1

复制

6 20

# 小鱼的游泳时间

---

例 2.10 (洛谷 P1425)

小鱼从 a 时 b 分一直游泳到当天的 c 时 d 分，请你帮小鱼计算一下，它这天一共游了多少分钟呢？

**输入格式：**一行内输入 4 个整数，分别表示 a, b, c, d。

**输出格式：**一行内输出 2 个整数 e 和 f，用空格间隔，依次表示小鱼这天一共游了多少小时多少分钟。其中表示分钟的整数 f 应该小于 60。

**输入样例：**12 50 19 10

**输出样例：**6 20

# 小鱼的游泳时间

## 例 2.10 (洛谷 P1425)

小鱼从 a 时 b 分一直游泳到当天的 c 时 d 分，这天游了几分钟？

可以把这两个时间统一转换成距离 0 点 0 分经过了多少分钟，然后相减就可以得到两个时间的分钟差，经过简单的整除和取余就可以得到答案。

```
#include<cstdio>
using namespace std;
int main() {
    int a, b, c, d, e, f, delta;
    scanf("%d%d%d%d", &a, &b, &c, &d);
    delta = (60 * c + d) - (60 * a + b);
    e = delta / 60;
    f = delta % 60;
    printf("%d %d", e, f);
    return 0;
}
```

# 成绩

例 2.11 (洛谷 P3954, NOIP2017 普及组)

某门课程计分方法是：总成绩 = 作业成绩  $\times$  20% + 小测成绩  $\times$  30% + 期末考试成绩  $\times$  50%。现在已经知道各项得分，求总成绩。

解法 1：直接加权计算。

`int(……)` 是指不是整数的内容强制转换为 `int` 类型，舍掉小数位数只保留整数。`double(……)` 可以强制转换为 `double`。

为什么最后加上了一个 0.5？是因为浮点误差（例如 89.9999）。

```
int a, b, c;
scanf("%d%d%d", &a, &b, &c);
printf("%d", int(a * 0.2 + b * 0.3 + c * 0.5 + 0.5));
return 0;
```

# 成绩

解法 2：注意“A,B,C 都是 10 的整数倍”。

每一小项经过加权，即使有乘上一个小数，但是结果一定是整数。

可以把  $*0.2$  变成  $*2/10$ ，使程序不接触浮点数，不必考虑浮点误差。

```
printf("%d", a * 2 / 10 + b * 3 / 10 + c * 5 / 10);
```

看题的时候不要遗漏这些有用的信息

# 提交评测与错误自查

我的程序通过不了，怎么办呢？

请翻至课本 P28，课件内容请参阅附录 A



# 课后习题与实验

学而时习之，不亦说乎。学而不思则罔，思而不学则殆。——孔子

请翻至课本 P30

# 判断练习

---

判断下列数据需要哪种类型的变量来存储：

A

3.1415926

1013

## 填空练习

---

请观察下列输出语句，并说出输出结果。

```
printf("%d%d", 12, 34);
```

```
printf("%d %d", 12, 34);
```

```
printf("%5d %5d", 12, 34);
```

```
printf("%d%.3f", 12, 34.0);
```

```
printf("%c", 65);
```

# 总结

---

`double`: 用于存放小数的变量类型。

`char`: 用于存放字符的变量类型。ASCII将每个字母和数字对应。

`cin>>a`: 用于读入输入的信息给表示变量 `a`。

一道题目如果按照题意不好解决，不妨想想能不能换一种数据处理的方式，比如把数字当成字符。

# 总结

---

占位符：scanf/printf

用于来格式化输入输出。需要牢记几个重要占位符，并区分。

算法竞赛基本格式：

每个题目都有固定的输入输出格式，需要认真阅读。

数据类型强制转换：

将其他类型的数据类型转换为需要的数据类型。

常见评测状态：

是对选手作答结果的反馈。

可以根据反馈的结果对代码进行修改。

# 作业

## 习题 2.1

编写程序解决应用的问题（具体请见课本 P30）。

问题和上一章的习题是完全一样的，只是要求题目中出现所有的数字都从键盘中输入（使用强调的数字），而不是直接写在程序中。然后手工计算，并验证答案。

## 习题 2.2：三角形面积（洛谷 P5708）

一个三角形的三边长分别是  $a$ 、 $b$ 、 $c$ ，那么它的面积为  $\sqrt{p(p-a)(p-b)(p-c)}$ ，其中  $p = \frac{(a+b+c)}{2}$ 。

输入三个数字，计算三角形的面积，四舍五入精确到 1 位小数。

提示：若故意输出不合法边长，比如两边和小于第三边，会怎样？

# 作业

## 习题 2.3:

尝试将本章例题中使用cin/cout输入输出的程序，改成使用scanf/printf输入输出，并测试。

## 习题 2.4:

阅读下列程序，猜测结果，并上机验证。试分析原因。

```
#include<cstdio>
using namespace std;
int main() {
    float a = 0.1;
    printf("%d", int(2 - a * a * 100));
    // printf("%.10f", 2 - a * a * 100);
    return 0;
}
```

# 作业

---

## 习题 2.5 小玉买文具 (洛谷 P1421)

小玉到文具店里买尽量多的签字笔。

一只签字笔的价格是 1 元 9 角，而班主任给小玉的钱是  $a$  元  $b$  角 ( $a \leq 10000, 0 \leq b \leq 9$ )

小玉想知道，她最多能买多少只签字笔呢。

## 习题 2.6 (选做) Apples Prologue (洛谷 P5709)

八尾勇喜欢吃苹果。她现在有  $m$  ( $m \leq 100$ ) 个苹果，吃完一个苹果需要花费  $t$  ( $t \leq 100$ ) 分钟，吃完一个后立刻开始吃下一个。

现在时间过去了  $s$  ( $s \leq 10000$ ) 分钟，请问还有几个完整的苹果？



## 参考阅读材料

---

以下的内容限于课件篇幅未能详细阐述。如果学有余力，可自行翻阅课本作为扩展学习。

- P18 例 2.3：强制类型转换
- P19 例 2.4：对浮点数四舍五入，和运行时错误
- P27 例 2.12：有一点复杂的顺序结构程序设计
- 习题 2.7。