计算机程序设计基础

Programming Fundamentals

韩文弢

清华大学计算机系

第二章 简单程序设计

1. 顺序结构程序设计
2. 选择结构程序设计

2.1 顺序结构程序设计

顺序结构程序设计

- ☺ 语句一条接一条地执行，没有分支、跳转等结构
- ☺ 最简单的一种程序结构

2.1.1 公式计算

问题描述：

输入一个角度 α，计算下列 y 的值：

$$y = \sqrt{\frac{1-\cos\alpha}{2}}$$

问题分析：

- 需要哪些数据？
  - 定义两个**double**类型的变量 **alpha** 和 **y**。
- 数学函数调用：余弦、平方根函数
  - **#include <cmath>**
  - **cos，sqrt**

```
double cos( double x );
```

- 功能：计算 x 的余弦值
- 说明：x 是弧度值。

$$\text{弧度} = \text{角度} * \pi / 180^{o}$$

```
#include  <iostream>
#include <iomanip>
#include <cmath>
using namespace std;
int  main( )
{
        double   alpha,  y;
        cout << "请输入一个角度：  ";
        cin >> alpha;
        y = sqrt((1 - cos(alpha / 180 * 3.14159)) / 2.0);
        cout << "y = " << fixed << setprecision(2) << y;
        return( 0);
}
```

一次运行结果

```
请输入一个角度：30
 y =  0.26
```

2.1.2 进制转换

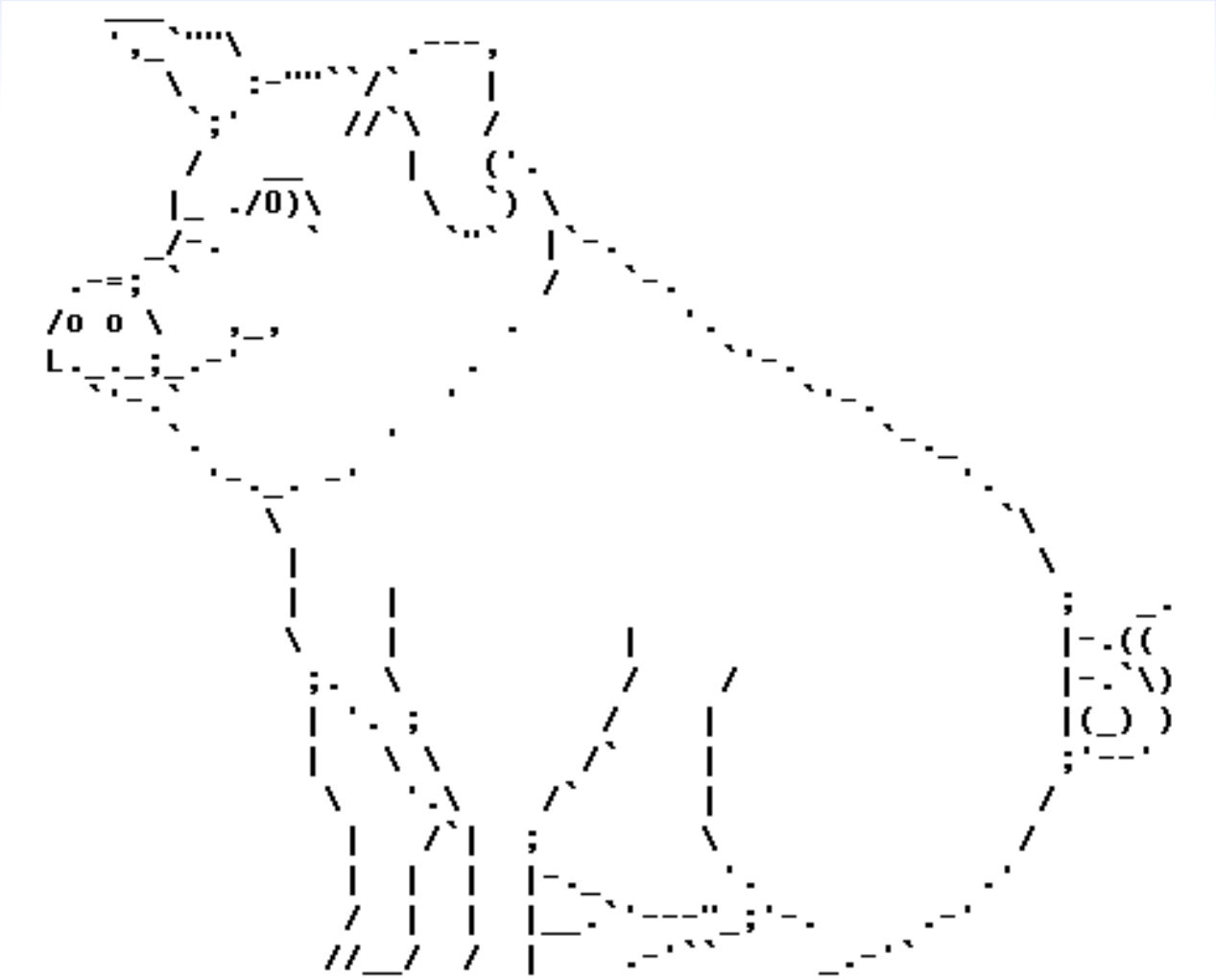
问题描述：

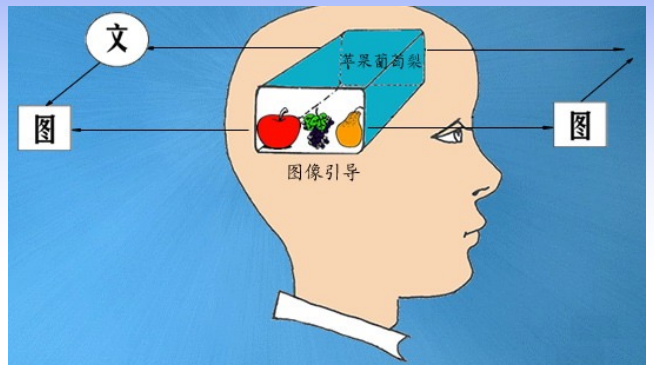
输入一个十进制整数，然后以字符形式输出它的十六进制形式。说明：假设该整数所对应的十六进制形式有两位数字，且每位数字的值在'A'~'F'之间。

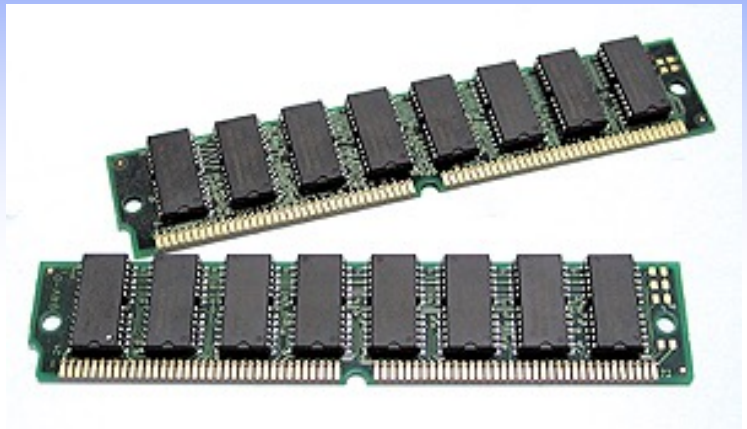
例如：假设输入为186，则相应的输出为：十六进制：0xBA。

字符类型

- 字符类型
  - 一个字符型数据只占用一个字节的空间。
  - 双重属性：**整数属性**和**字符属性**。
- 整数属性
  - 字符类型即**单字节的整数类型**。
- 字符属性
  - 数据值即为相应字符的 ASCII 码。







- 存放方式？存放内容？
  - ☺ **二进制、ASCII值（如'A'，65）**
- 使用方式？
  - ☺ **当成整数或字符**

L　H	000	001	010	011	100	101	110	111
0000	NUL		(space)	0	@	P	`	p
0001			!	1	A	Q	a	q
0010			"	2	B	R	b	r
0011			#	3	C	S	c	s
0100			$	4	D	T	d	t
0101			%	5	E	U	e	u
0110			&	6	F	V	f	v
0111	(beep)		'	7	G	W	g	w
1000			(	8	H	X	h	x
1001	(tab)		)	9	I	Y	i	y
1010	LF		*	:	J	Z	j	z
1011	VT		+	;	K	[	k	{
1100	FF		,	<	L	\	l	|
1101	CR		-	=	M	]	m	}
1110			.	>	N	↑	n	~
1111			/	?	O	←	o	DEL

注：H 表示高 3 位，L 表示低 4 位。

最多描述多少个字符？

字符常量

用单引号括起来的一个字符，如'a'、'A'、'#'。

```
('F' - 'A') + 'd' = 'i'
```

转义字符表

Why 转义字符？

符号	ASCII值	含义
\a	007	响铃
\b	008	退格
\n	010	换行
\r	013	回车
\t	009	水平Tab键
\v	011	竖直Tab键
\'	039	单引号 '
\"	034	双引号 "
\\	092	反斜杆 \
\ooo	—	八进制表示的字符
\xhh	—	十六进制表示的字符

如：'\x41'

```
cout << "\a\a";
cout << "你\t好\n你有";
cout << "\"计算机语言与程序设计\"这本书吗?";
```

```
你    好
你有"计算机语言与程序设计"这本书吗?
```

清华男生为防偷看用电脑编码写明信片

char s[100] = {214, 247, 210, 170, 190, 205, 202, 199, 207, 235, 202, 212, 202, 212, 184, 248, 212, 219, 195, 199, 188, 196, 208, 197, 202, 177, 181, 216, 214, 183, 191, 201, 210, 212, 208, 180, 181, 195, 182, 224, 188, 242, 194, 212, 161, 173, 161, 173};

```
#include <iostream>
using namespace std;
int main()
{
      int value;
      int v1, v2;
      char c1, c2;

      cout << "请输入一个整数:";
      cin >> value;

      v1 = value / 16;
      v2 = value % 16;
      c1 = v1 - 10 + 'A';
      c2 = v2 - 10 + 'A';


      cout << "0x" << c1 << c2 << endl;
}
```

```
请输入一个整数:186

0xBA
```

2.2 选择结构程序设计

Where to go?!

假币问题

有五枚硬币，但其中有一枚是假币。已知假币比真币要轻。现有一架天平，请问：最多需要称几次就能把假币找出来？

答案：二次即可。

任取4枚硬币，放在天平上，一边2枚：

- *if* 左边的2枚比较轻，则把它们再称一次，轻者即为假币；
- 否则，*if* 右边的2枚比较轻，则把它们再称一次即可；
- 否则，剩下的那枚硬币是假币。

2.2.1 关系运算符和关系表达式

所谓的“**关系运算**”实际上是“**比较运算**”。

C/C++语言提供了 6 种关系运算符：

(1) <　　（小于）
(2) <=　　（小于或等于）
(3) >　　（大于）
(4) >=　　（大于或等于）

优先级相同（高）

(5) ==　　（等于）
(6) !=　　（不等于）

优先级相同（低）

例如：a > b == c　　等价于 (a > b) == c

a == b < c　　等价于 a == (b < c)

优先级高

算术运算符：+、－、*、/、%

关系运算符：>、<、==、>=、<=、!=

优先级低

赋值运算符：＝

例如：c > a + b　　等价于 c > (a + b)

a = b > c　　等价于 a = (b > c)

- 关系表达式
  - **用关系运算符将两个操作数连接起来的式子**
- 关系表达式的值
  - **逻辑值“真”或“假”，即布尔类型。**

布尔类型

假做真时真亦假，无为有处有还无 …

布尔类型的取值

☺ 布尔类型变量的取值为：true 或false，用来表示逻辑运算的结果，即1或0。

```
bool RealMonkey = true;

bool RealMoney;

RealMoney = false;
```

```
int y = 5;              // y初始化为5
bool x;
x = (y < 4);            // x等于false，即0
x = (y >= 5);           // x等于true，即1

int flower;
x = (flower != flower); // x等于false
```

- 关系表达式的局限性
  - **只能描述一个条件；**
  - **如果有多个条件，例如，在 [10, 20] 的范围内，如何表达？**

2.2.2 逻辑运算符和逻辑表达式

用逻辑运算符将关系表达式或逻辑量连接起来的式子，就是逻辑表达式。

C/C++语言提供了 3 种逻辑运算符，包括：

(1) &&　　（逻辑与，双目运算符）
(2) ||　　（逻辑或，双目运算符）
(3) !　　（逻辑非，单目运算符）

a && b

a \ b	真	假
真	真	假
假	假	假

a || b

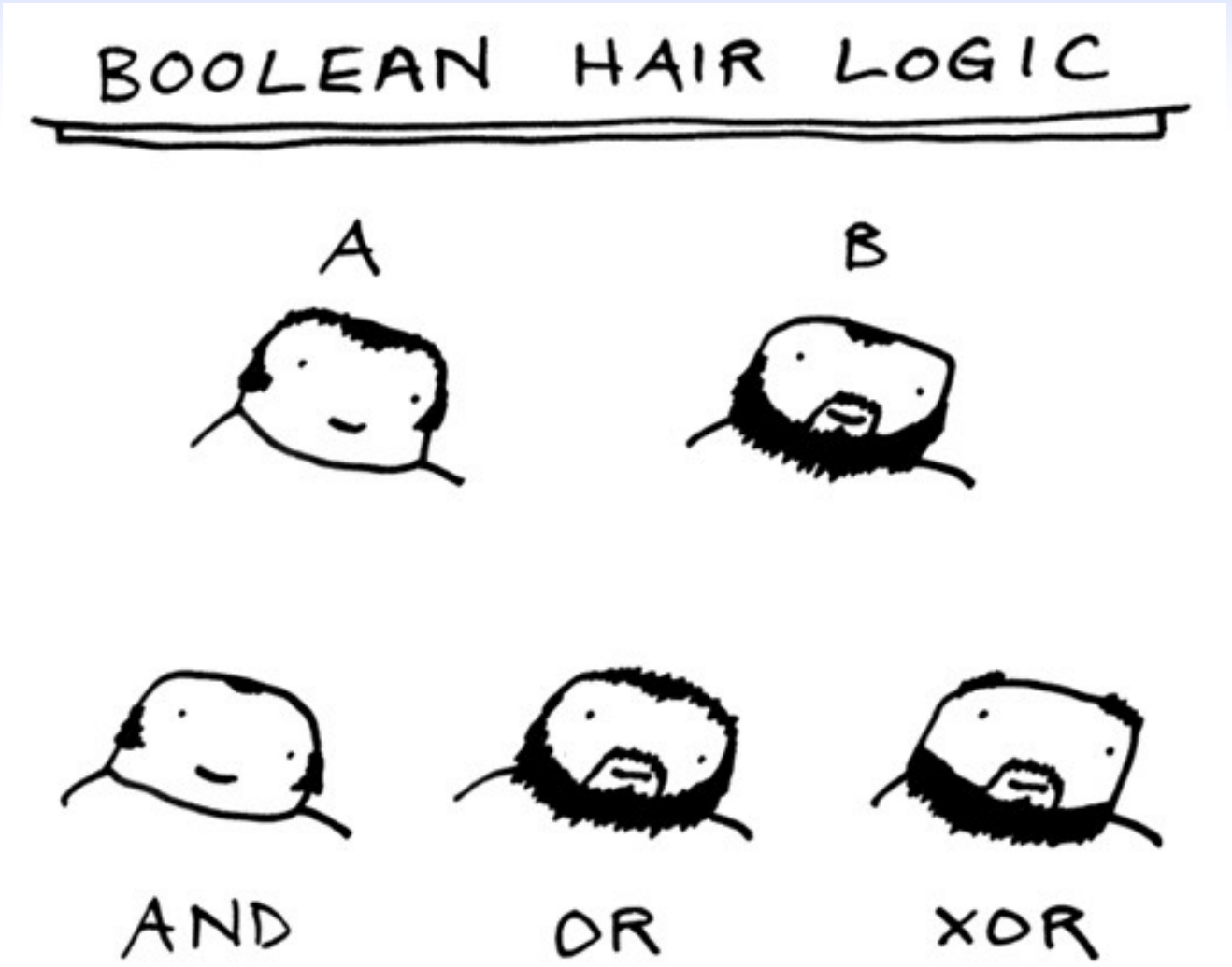
a \ b	真	假
真	真	真
假	真	假

!a

a	真	假
!a	假	真

真和假

- C++ 中，
  - 真是 true (1)，假是 false (0)
  - 非零是真，零是假
- 思考以下表达式的值：
  - !0　!1　!7　!!7
  - 1　0　0　1


BOOLEAN HAIR LOGIC
A
B
AND
OR
XOR

逻辑操作符的短路设计

- 对于逻辑与操作符 &&，如果已知它左边的值是假，则直接返回假，不再计算它右边的值。
- 对于逻辑或操作符 ||，如果已知它左边的值是真，则直接返回真，不再计算它右边的值。
- 对于右边的表达式有副作用的情况，短路发生时会产生区别，右边的表达式没有被计算，因此也不会发生副作用。

2.2.3 if 语句

if 语句是一种分支语句，它用来判定所给出的条件是否满足，然后根据判定的结果（真或假）来决定执行相应的操作。

if语句的形式之一

```
if（表达式）语句1；        // 语句1，只一句
```

如果表达式为真，执行语句1；否则什么都不做

例如：

```
if (x  >  y)    cout << "最大值是" << x << endl;

if (temperature  >  38)
        cout << "You have a fever. \n";
        cout << "Go see the doctor \n";
```

有点问题吧？

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

引入复合语句：若表达式为真，执行复合语句当中的每一条语句；否则什么都不做；

```
if(temperature > 38)
{
    cout << "You have a fever. \n";
    cout << "Go see the doctor \n";
}
```

if语句的形式之二

```
if（表达式）语句1；
else   语句2；
```

如果表达式为真，执行语句1；否则执行语句2

其中，语句1和语句2可以是单条语句，也可以是复合语句。

一个关于码农的段子

老婆给当程序员的老公打电话：“下班顺路买一斤（10个）包子带回来，如果看到卖西瓜的，买一个。”

当晚，程序员老公手捧着一个包子进了家门。。。

老婆怒道：“你怎么就买了一个包子？！”老公答曰：“因为看到了卖西瓜的。”

一个关于码农的段子(2)

```
// what the wife thought
BuyBaoZi(10);
if(看到卖西瓜的) {
    BuyWatermelon(1);
}
```

```
// what the husband thought
if(看到卖西瓜的)     BuyBaoZi(1);
else BuyBaoZi(10);
```

计算绝对值

问题描述：

计算x的绝对值|x|，把结果保存在abs变量中。

方案1

```
if(x >= 0) abs = x;
if(x < 0 ) abs = -x;
```

方案2

```
abs = x;
if(x < 0) abs = -x;
```

方案3

```
if(x >= 0) abs = x;
else abs = -x;
```

哪一个正确？

if语句的形式之三

```
if（表达式1）语句1;
else  if（表达式2）  语句2;
else  if（表达式3）  语句3;
……
else  if（表达式m）  语句m;
else  语句m+1;
```

层叠的 if 语句

如果表达式1为真，执行语句1；否则如果表达式2为真，执行语句2；否则如果表达式3为真，执行语句3；否则,…，如果表达式m为真，执行语句m，否则执行语句m+1。

寻找死代码

```
1       if (speed >= 0)
2       {
3               cout << "You don't go backward.\n";
4               if (speed == -1) {
5                       cout << "Wait! I was wrong!";
6               }
7       }
8       else if (speed > 0) {
9               cout << "You go forward.\n";
10      }
11      else if (speed < 0) {
12              cout << "You go backward.\n";
13      }
14      else {
15              cout << "What did you do?!\n";
16      }
```

世上倒数第二个C/C++ Bug

```
if ( 0 <= x <= 10 )
{
    cout << "x is between 0 and 10. \n ";
}
```

世上最后一个C/C++ Bug

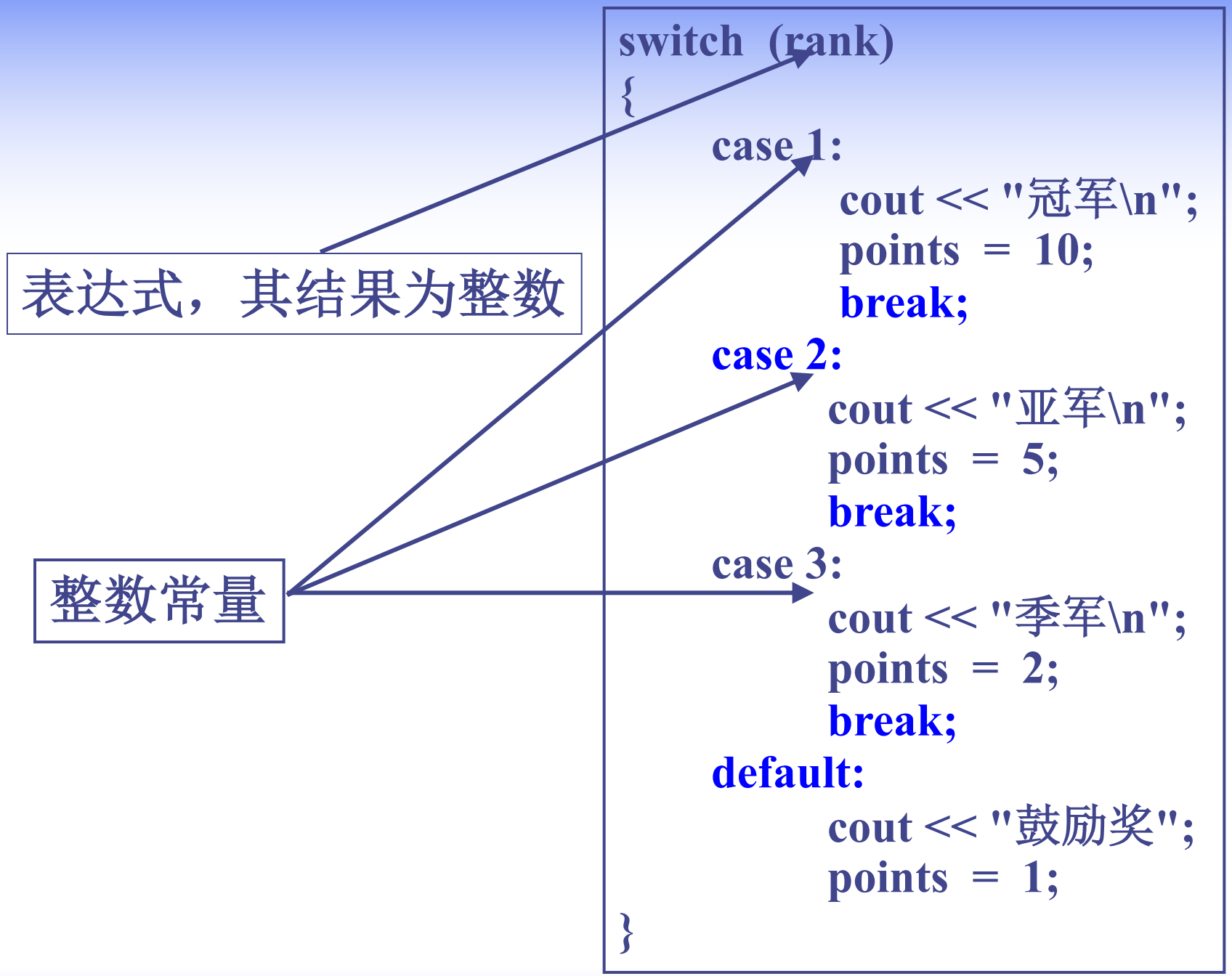
```
status = check_radar ( ) ;
if (status = true)
{
        launch_nuclear_missiles ( ) ;
}
```

July 28, 1962 -- Mariner I space probe. A bug in the flight software for the Mariner 1 causes the rocket to divert from its intended path on launch. Mission control destroys the rocket over the Atlantic Ocean. The investigation into the accident discovers that a formula written on paper in pencil was improperly transcribed into computer code, causing the computer to miscalculate the rocket's trajectory.

2.2.4 switch 语句

层叠的 if 语句：

```
if (rank == 1){
        cout << "冠军\n";
        points = 10;
}
else if (rank == 2){
        cout << "亚军\n";
        points = 5;
}
else if (rank == 3) {
        cout << "季军\n";
        points = 2;
}
else {
        cout << "鼓励奖";
        points = 1;
}
```


表达式，其结果为整数
整数常量
switch (rank)
{
case 1:
cout << "冠军\n";
points = 10;
break;
case 2:
cout << "亚军\n";
points = 5;
break;
case 3:
cout << "季军\n";
points = 2;
break;
default:
cout << "鼓励奖";
points = 1;
}

2.2.5 进制转换2

问题描述：

编写一个程序，输入一个255以内的正整数，然后以字符形式输出它的十六进制形式。例如：假设输入为140，则相应的输出为：十六进制：0x8C。（讨论）

1. 该十六进制数有几位？
2. 每一位的取值范围？

问题分析：

1. 该数小于255，则相应的十六进制数最多为2位（可能只有1位）；

2. 对于每一位十六进制数，它既可能大于也可能小于10，若大于10，需要转换。

1. 该十六进制数只有1位
   1.1 小于10
   1.2 大于或等于10
2. 该十六进制数有2位
   2.1 处理第1位（大于或小于10）
   2.2 处理第2位（大于或小于10）

```
#include <iostream>
using namespace std;
int main()
{
    int value, v1, v2;
    char c1, c2;

    cout << "请输入一个整数:";
    cin >> value;

    v1 = value / 16;
    v2 = value % 16;

    if(v1 < 10)    c1 = v1 + '0';
    else  c1 = v1 - 10 + 'A';

    if(v2 < 10)  c2 = v2 + '0';
    else  c2 = v2 - 10 + 'A';

    cout << "十六进制: 0x" << c1 << c2 << endl;
}
```

几次运行结果

```
请输入一个整数:9
0x09
```

```
请输入一个整数:12
0x0C
```

```
请输入一个整数:140
十六进制: 0x8C
```

2.2.6 今天星期几？

问题描述：

我们经常想知道历史上一些重要的日子是星期几，或想知道将来的某一天是星期几，那么，怎样计算任意一天是星期几呢？

问题分析：

根据历法，可按以下方法计算某年某月某日是星期几。首先，根据下列公式计算*S*：

$$S = y - 1 + \left\lfloor \frac{y-1}{4} \right\rfloor - \left\lfloor \frac{y-1}{100} \right\rfloor + \left\lfloor \frac{y-1}{400} \right\rfloor + C$$

这里*y*是公元的年数，*C*是从这一年的元旦算起到这天为止（包括这天在内）的天数，⌊ ⌋表示向下取整。

得到*S*后，再用7除，取余数。余数是几就是星期几。若余数为零，则为星期日。

例如：计算1949年10月1日是星期几？

$$S = y - 1 + \left\lfloor \frac{y-1}{4} \right\rfloor - \left\lfloor \frac{y-1}{100} \right\rfloor + \left\lfloor \frac{y-1}{400} \right\rfloor + C$$

C＝31＋28＋31＋30＋31＋30＋31＋31＋30＋1

＝274

$$S = 1949 - 1 + \left\lfloor \frac{1949-1}{4} \right\rfloor - \left\lfloor \frac{1949-1}{100} \right\rfloor + \left\lfloor \frac{1949-1}{400} \right\rfloor + C$$

＝1948＋487－19＋4＋274

＝2694

＝384×7＋6

所以，1949年10月1日是星期六。

问题分析：

本问题的关键在于如何计算*S*，而计算*S*的关键在于如何计算*C*。不妨令这一天为y年、m月、d日，则本问题的关键在于如何计算从y年的1月1日开始，到m月d日为止的天数。

基本思路：

C ＝ 前m–1个月的总天数 ＋ d。

有两点需要注意：

- 每个月的天数是固定的：一、三、五、七、八、十和十二月份是31天，四、六、九和十一月份是30天，二月份一般是28天；
- 如果是闰年，则二月份是29天。

一种方法：

- 把m分为十二种情形来讨论，即m=1、m=2、m=3、…、m=12，事先计算出在每一种情形下，前m–1个月的总天数；
- m = 1：$C = 0 + d$；
- m = 2：$C = 31 + d$；
- m = 3：$C = 31 + 28 + d = 59 + d$；
- m = 4：$C = 31 + 28 + 31 + d = 90 + d$；
- ……

可能的源程序片断

```
switch ( m )
{
        case 1:   C = d;          break;
        case 2:   C = 31 + d;     break;
        case 3:   C = 59 + d;     break;
        case 4:   C = 90 + d;     break;
        ……
}
```

思路分析：

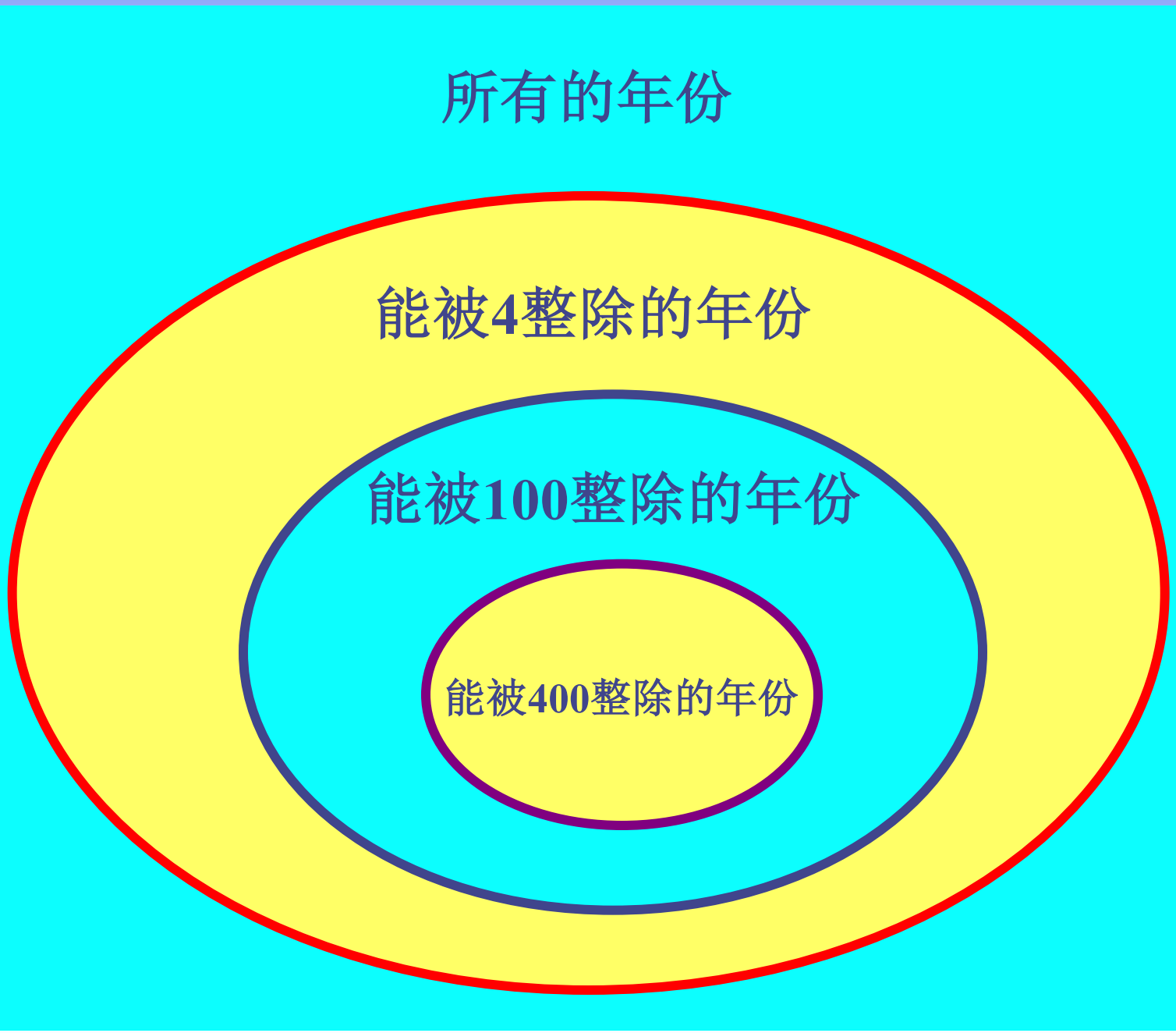
计算前m–1个月的总天数，实际上是统计在前m–1个月当中，有多少个月份是31天，多少个月份是30天，以及二月份是28天还是29天。不妨令day31、day30和day28分别表示在前m–1个月中，31天、30天和28天的月份个数，它们的具体数目如下(找规律)：

m=	1	2	3	4	5	6	7	8	9	10	11	12
day31	0	1	1	2	2	3	3	4	5	5	6	6
day30	0	0	0	0	1	1	2	2	2	3	3	4
day28	0	0	1	1	1	1	1	1	1	1	1	1

```
if (m  ==  1)    day31 = day30 = day28 = 0;
else  if (m  ==  2)
{
      day31 = 1;     day30 = day28 = 0;
}
else  if ( m  <=  8)
{
      day31  =  m / 2;     day28 = 1;
      day30  =  m -1 - day31 - day28;
}
else
{
      day31  =  (m + 1) / 2;     day28 = 1;
      day30  =  m -1 - day31 - day28;
}
C  =  day31 * 31 + day28 * 28 + day30 * 30 + d;   // 闰年?
```

闰年的规则：

① 能被4整除，但不能被100整除的年份都是闰年，如1996年、2004年；② 能被100整除，又能被400整除的年份是闰年，如1600年、2000年；③ 除此之外的所有年份都不是闰年。


所有的年份
能被4整除的年份
能被100整除的年份
能被400整除的年份

数据结构：

需要一个整型变量 year，记录年份。

规则分析：

① 能被4整除，但不能被100整除的年份都是闰年；写成逻辑表达式就是：

(year % 4 == 0 && year % 100 != 0)

② 能被100整除，又能被400整除的年份是闰年；写成关系表达式就是：

(year % 400 == 0)

```
int main( )
{
    int y, m, d;
    int S, C, day31, day30, day28;

    cin >> y >> m >> d;
    if (m == 1)     day31 = day30 = day28 = 0;
    else if (m == 2)
    {
        day31 = 1;      day30 = day28 = 0;
    }
    else  if ( m  <=  8)
    {
        day31  =  m / 2;      day28 = 1;
        day30  =  m -1 - day31 - day28;
    }
    else
    {
        day31  =  (m + 1) / 2;      day28 = 1;
        day30  =  m -1 - day31 - day28;
    }
```

```
    C  =  day31 * 31 + day28 * 28 + day30 * 30 + d;

    if((y % 4 == 0  &&  y % 100 != 0)
       || (y % 400  == 0))
    {
        if(m > 2) C = C + 1;
    }
    S = y - 1 + (y-1)/4 - (y-1)/100 + (y-1)/400 + C;

    cout << y << "年" << m << "月" << d
         << "日是星期 " << S%7 << endl;

    return 0;
}
```

本讲小结

- 顺序结构
- 数学库
- 字符类型
- 布尔类型和关系运算、逻辑运算
- 选择结构（分支、条件）

- C++ 参考手册：https://zh.cppreference.com/