

计算机程序设计基础

Programming Fundamentals

韩文弢

清华大学计算机系



教学安排

◆ 教学对象

非计算机专业本科生

◆ 教学目标

- ◆ **掌握一种编程工具（C++语言）**
- ◆ **掌握程序设计的基本思路和方法，提高分析问题、解决问题的能力**
- ◆ **学习面向对象程序设计思想**

◆ 教学方法

- ◆ 课堂讲解：基本概念、C++语言的语法、语句，编程解题的基本思路、方法。鼓励创新、鼓励上台讲解自己的方法；
- ◆ 上机实践：**实践性极强的课程**，将有大量的编程练习。每天有固定上机时间，完成作业、助教答疑。
- ◆ 每天13:30-15:05，东主楼9-224机房

◆ 考核方式（百分制成绩）：

上机作业 + 课程项目 + 课堂测试 + 附加
（ 30% ） （ 50% ） （ 20% ） （ +5% ）

◆ 严格遵守学术诚信，禁止抄袭代码。

◆ 课程站点

◆ <https://learn.tsinghua.edu.cn/>

◆ <https://oj.cs.tsinghua.edu.cn/>

主讲教师

韩文弢

- ◆ 单位：计算机系
- ◆ 研究方向：计算机系统
 - 大数据处理系统
 - 大规模机器学习系统
- ◆ 邮箱：hanwentao@tsinghua.edu.cn
- ◆ 学术主页：<https://pacman.cs.tsinghua.edu.cn/~hanwentao/>
- ◆ 办公室：东主楼 9-324



课程助教



孙桢波

sunzb20@mails.tsinghua.edu.cn



翟明书

dms22@mails.tsinghua.edu.cn

学编程到底学什么？

◆ 写作

- 自然语言
- 写作逻辑

◆ 编程

- 编程语言
- 计算思维（算法等）

计算机专业的编程相关课程

- ◆ 程序设计基础(FOP)：初步掌握一门编程语言(C++)，实现简单算法，体会计算思维——入门
- ◆ 面向对象程序设计基础(OOP)：学习面向对象编程的语言特性，学习体会设计思想——提高
- ◆ 程序设计训练(P&T)：学习一门新的编程语言(Rust、Java 或 Python)，体会编程语言的设计理念，锻炼解决问题的能力——融会贯通
- ◆ 软件工程(SE)：系统设计，项目管理，团队合作
- ◆ 操作系统(OS)：结合计算机系统结构的设计与实现，抽象概念

第一章 程序设计概述

1. 计算机与程序
2. 计算机程序设计
3. 一个典型的计算机程序
4. R进制

1.1 计算机与程序

这是一个计算机的时代



May 11th, 1997

Computer won world champion of chess

(Deep Blue)

(Garry Kasparov)



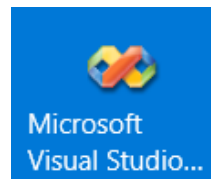
计算机 = 电脑？

计算机 + 软件 = 电脑

何为计算机程序？



QQ.exe



Visual Studio.exe



iexplore.exe



winamp.exe

1.2 计算机程序设计

✦ 输入：文字形式的问题描述

☺ “请问计算机， $1+2+\dots+100$ 等于多少”

☺ “请编写一个即时通信软件”

✦ 输出：可执行程序

☺ **Sum.exe**

☺ **NewQQ.exe**

- ◆ 程序员的工作：问题描述 → 可执行程序
- ◆ 如何编写程序？
 - 接受一个问题；
 - 分析这个问题；
 - 设想一种方法来解决这个问题；
 - 在计算机上，用某种**计算机语言**来描述该解决方案，即编码；
 - 测试和调试程序（**debug**）。

- 编程就象厨师烧菜，以红烧茄子为例；
- 原料（数据）：茄子750克、肉片50克、葱姜蒜50克、酱油、盐、糖、鸡粉适量，水淀粉适量。
- 制作过程（算法）：
 - ✓ 将茄子切成块，葱姜蒜切末待用；
 - ✓ 起油锅，待油热至八、九成时放入茄子，炸至茄子由硬变软时取出，将油沥干待用；
 - ✓ 另起锅，锅中放油三汤匙，油热后，先爆香葱姜蒜，下肉片炒散，烹入酱油，加入少量水和鸡粉，放入茄子、盐、糖，大火煮开后改用小火煮至茄子入味，最后用淀粉勾芡。
- **程序 = 数据结构 + 算法**

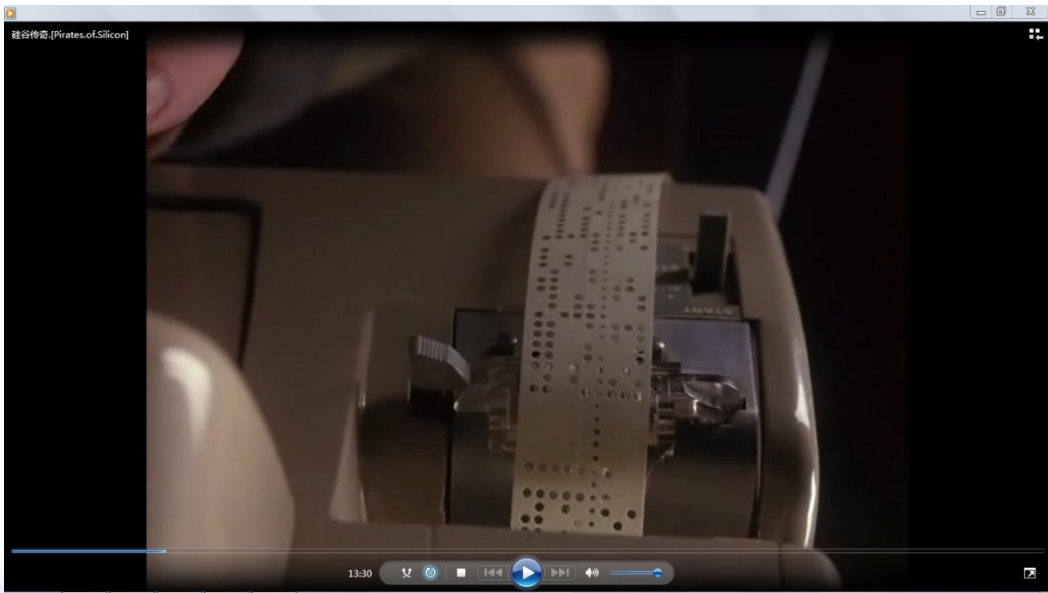
计算机语言

```
C:\Windows\system32\cmd.exe - debug qq.exe

C:\>cd temp
C:\Temp>debug qq.exe
-t
AX=0000 BX=001B CX=A350 DX=0000 SP=00B6 BP=0000 SI=0000 DI=0000
DS=17D1 FS=17D1 SS=17E1 CS=17E1 IP=0001  NU UP EI PL NZ NA PO NC
17E1:0001 1F POP DS
-t
AX=0000 BX=001B CX=A350 DX=0000 SP=00B8 BP=0000 SI=0000 DI=0000
DS=17E1 FS=17D1 SS=17E1 CS=17E1 IP=0002  NU UP EI PL NZ NA PO NC
17E1:0002 BA0E00 MOV DX,000E
-t
AX=0000 BX=001B CX=A350 DX=000E SP=00B8 BP=0000 SI=0000 DI=0000
DS=17E1 FS=17D1 SS=17E1 CS=17E1 IP=0005  NU UP EI PL NZ NA PO NC
17E1:0005 B409 MOV AH,09
-t
AX=0900 BX=001B CX=A350 DX=000E SP=00B8 BP=0000 SI=0000 DI=0000
DS=17E1 FS=17D1 SS=17E1 CS=17E1 IP=0007  NU UP EI PL NZ NA PO NC
17E1:0007 CD21 INT 21
-
```

机器指令
(直接在硬件执行)

汇编指令
(用符号表示操作码和数据)



◆ 高级语言

- 用一种更自然、更接近于人类语言习惯的符号形式来编写程序

◆ 发展历史

- 1950's : Fortran, Lisp
- 1960's : Cobol, Algol, APL
- 1970's : Basic, Pascal, C
- 1980's : Smalltalk, C++, Modula, Ada, Prolog
- 1990's : Java
- Now: Python, Perl, PHP, C#, JavaScript, Ruby...

1.3 一个典型的计算机程序

问题描述

**“编写一个程序，计算所有课程的
GPA (Grade Point Average) ”**

4分制 (0 - 4)

百分制分数	等级	grade point
90-100	A	4
80-89	B	3
70-79	C	2
60-69	D	1
60以下	F	0

计算方法

将每门课程的学分乘以学时，累加后除以总学时，得出平均分。

例如：某学生的五门课程的学时和成绩为：

A课程4个学时，成绩92（A）；

B课程3个学时，成绩80（B）；

C课程2个学时，成绩98（A）；

D课程6个学时，成绩70（C）；

E课程3个学时，成绩89（B）。

假设所有课程成绩均不低于C，且学时数相同。

$$\text{GPA} = (4 \times 4 + 3 \times 3 + 2 \times 4 + 6 \times 2 + 3 \times 3) / (4 + 3 + 2 + 6 + 3) = 3.00$$

基本思路

- 让用户输入成绩分别为A、B、C的课程数
 - 计算总分数（分子）
 - 计算总的课程数（分母）
 - 计算GPA
 - 在屏幕上显示计算结果
1. 变量的定义与使用
 2. 算术运算
 3. 输入和输出

1.3.1 变量的定义与使用

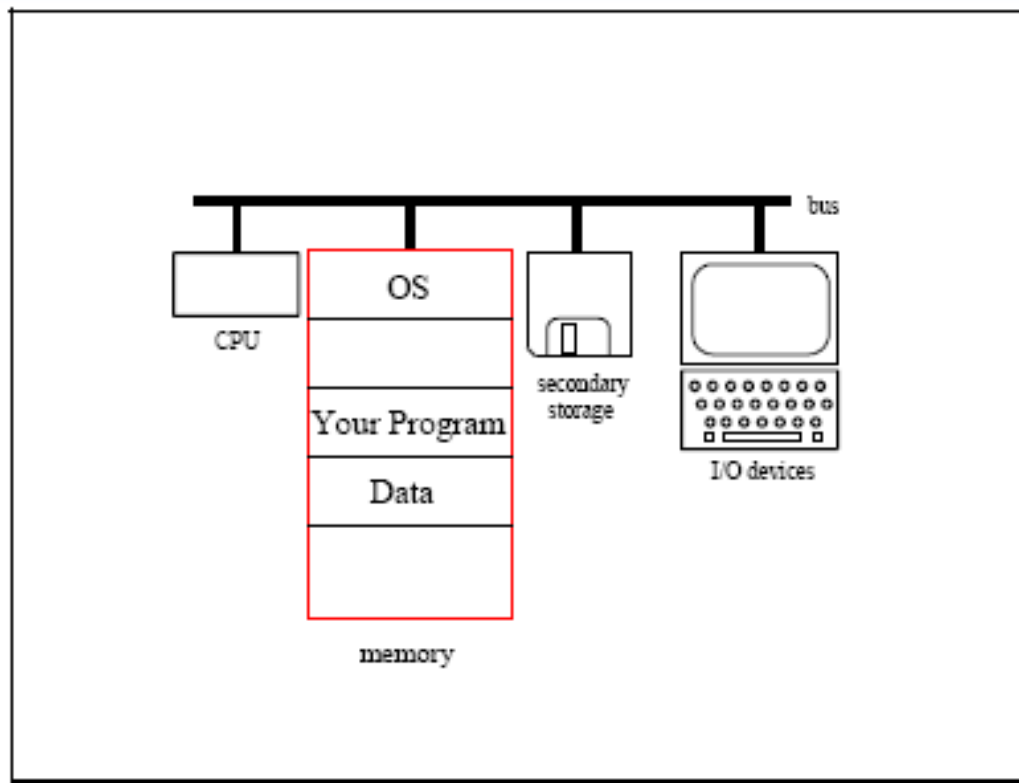
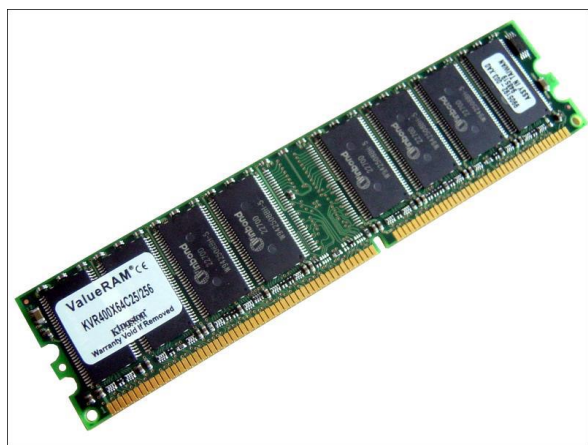
why变量？



茄子750克、肉片50克、
葱姜蒜50克、酱油、盐、
糖、鸡粉适量，水淀粉
适量。

有何共同点？

代码和数据 都是存放在 内存中



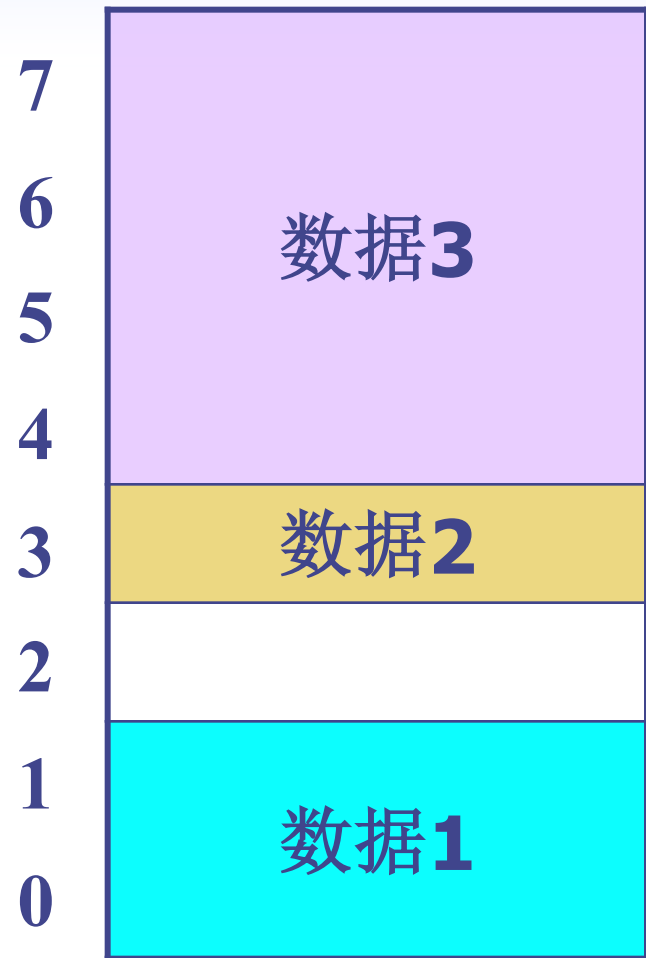
内存的工作原理

- 一个内存中包含有许多存储单元，每个单元可以存放一个适当单位的信息（如：8个bit，即一个字节，byte）；
- 全部存储单元按一定顺序编号，这种编号称为存储器的地址。对各个存储单元的读写操作就是通过它们的地址来进行的。

7	
6	
5	0 1 0 0 0 0 1 1
4	
3	
2	0 0 1 1 0 0 0 0
1	0 0 1 1 0 1 0 0
0	0 0 1 1 0 0 1 0

不同的数据，可能需要
不同长度的存储空间，
怎么办？

如：1、300、70000



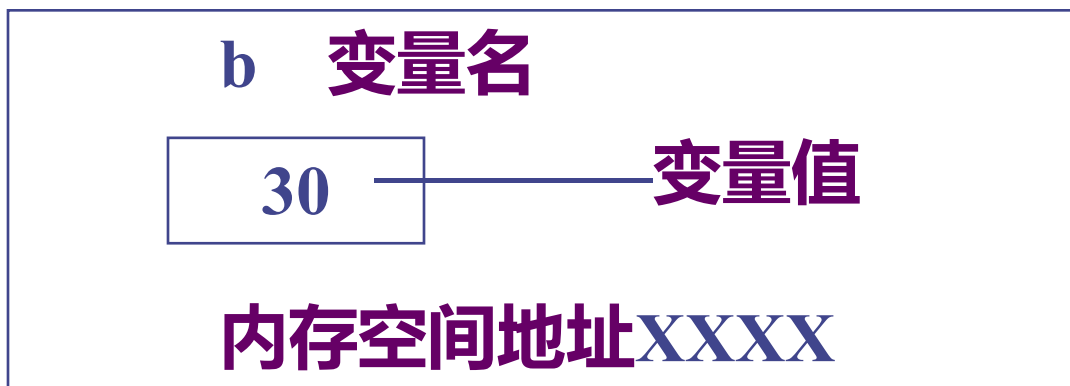
解决之道 —— 数据类型

- 把所有的数据归纳为有限的几种类型；
- 同一种类型的数据具有相同的长度，占用相同大小的内存空间；
- 每一种类型的数据依然是以二进制的形式存放在内存当中；
- 在访问一个数据时，根据它在内存的起始地址和类型来确定它所占用的存储单元。

- C++语言的四种基本类型：
 - 字符类型：用 **char** 来表示；
 - 整数类型：用 **int** 来表示；
 - 布尔类型：用**bool** 来表示；
 - 双精度浮点类型：用 **double** 来表示。
- 此外，还有一些类型修饰符：**short**、**long**、**signed**、**unsigned**。

变量的基本概念

变量：其值可变的量。



通过 **b** 可以找到相应的存储空间地址**XXXX**，
从而对该变量的值进行访问和修改。

变量的命名规则

- 仅包含字母、数字和下划线（ ‘_’ ）；
- 第一个字符必须为字母或下划线；
- 不能使用C++语言保留的“关键字”来作为变量名，如int, double等；
- 变量名是大小写有关的，例如：sum和SUM是两个不同的变量名。

● 合法的标识符

_BandNames
F4

TFBoys
Twins

S_H_E

◆ 非法的标识符

Magic1+1
Y-Star
Angel Girl

183club **S.H.E**
Robynn&kendy

中文变量名

变量的定义

数据类型 变量1, 变量2, ..., 变量n;

例如:

```
int nA, nB, nC;  
double totalCourses, totalPoints, gpa;
```

整数类型

整数类型可分为：基本型、短整型和长整型三种。

1. 基本型： `int`（4字节）；
2. 短整型： `short int`，或 `short`（2字节）；
3. 长整型： `long int`，或 `long`（4字节）；

无符号整数类型： `unsigned int`， `unsigned short` 和 `unsigned long`。

整型数据的长度及取值范围

数据类型	字节数	比特数	取值范围
int	4	32	$-2^{31} \sim (2^{31} - 1)$
short	2	16	$-2^{15} \sim (2^{15} - 1)$
long	4 (8)	32	$-2^{31} \sim (2^{31} - 1)$
unsigned int	4	32	$0 \sim (2^{32} - 1)$
unsigned short	2	16	$0 \sim (2^{16} - 1)$
unsigned long	4 (8)	32	$0 \sim (2^{32} - 1)$

实数类型

实数类型（浮点类型）：分为单精度浮点类型（**float**）、双精度浮点类型（**double**）和长双精度浮点类型（**long double**）三种。

实型数据的存放形式：小数部分+指数部分



实数类型(2)

IEEE 754 Standard

单精度和双精度浮点类型的表示:

S	EEEEEEEE	FFFFFFFFFFFFFFFFFFFFFFFF
0	1	8 9 31
S	EEEEEEEEEEEE	FF
0	1	11 12 63

S为符号位, E为指数位, F为小数位。

以单精度为例: 其值为 $(-1)^S * 2^{(E-127)} * (1.F)$

实数类型(3)

例如：把十进制数6.5转换为浮点数

1、进制转换

$$(6.5)_{10} = (110.1)_2$$

2、规格化

$$(110.1)_2 = (1.101) * 2^2 = (1.101) * 2^{10} \text{ // } 10 \text{ 是二进制}$$

3、计算指数位

问题？

$$10 + 01111111 = 10000001$$

4、 0 10000001 1010000000000000000000000000

小数部分占的位数越多，数据的有效数字越多，精度越高；指数部分占的位数越多，则能表示的数值范围越大。

各种实型数据

类型	字节数	有效数字	数值范围
float	4	6~7	$10^{-38} \sim 10^{38}$
double	8	15~16	$10^{-308} \sim 10^{308}$
long double	16 (8, 10)	18~19	$10^{-4932} \sim 10^{4932}$

1.3.2 算术运算

算术运算符

+：加法运算符，如 $3 + 5$ ；

-：减法运算符，如 $5 - 2$ ；

*****：乘法运算符，如 $3 * 5$ ；

/：除法运算符。如 $5 / 3$ 。**两个整数相除，结果为整数，小数部分被舍去；**

%：模运算符，或称求余运算符，**%两侧均为整型数据，如 $7 \% 4$ 。**

一些基本概念

- 表达式：有“值”的式子，通常由一些变量、常量、函数调用和运算符组合。
- 算术表达式：用算术运算符和括号将运算对象（也称操作数）连接起来的式子。
 - 运算符的优先级：在表达式求值时，先按运算符的优先级的高低次序执行，如先乘除后加减
 - 运算符的结合性：若一个运算对象两侧的运算符的优先级相等，则按规定的“结合方向”处理。算术运算符的结合方向为从左到右，即“左结合性”
 - 分不清优先级和结合性：加括号。

- 类型转换：把一种类型数据转换成另一种类型
 - 赋值转换（系统自动进行）：赋值运算符两侧的类型不一致，如： `int x = 3.14;`
 - 运算转换（系统自动进行）：运算符带有不同数据类型的运算对象，如： `5 * 100.0;`
 - 强制转换（程序员指定）：程序员使用强制类型转换运算符，如： `rate = (double)earn / capital。`

考察下列程序，计算 num4 的值：

```
int    num1,    num2;  
  
double num3,    num4;  
  
num1 = 5.999;  
  
num2 = -2;  
  
num3 = 5;  
  
num4 = num1/num2 + (int) (num3* (num3/2) ) ;
```

num4的值为： **10.0**

星期几？

小虎刚刚大学毕业参加工作，作为上班一族，他对每周七天的感觉是：

星期一，走向深渊；

星期二，路漫漫；

星期三，夜茫茫；

星期四，黎明前的黑暗；

星期五，归心似箭；

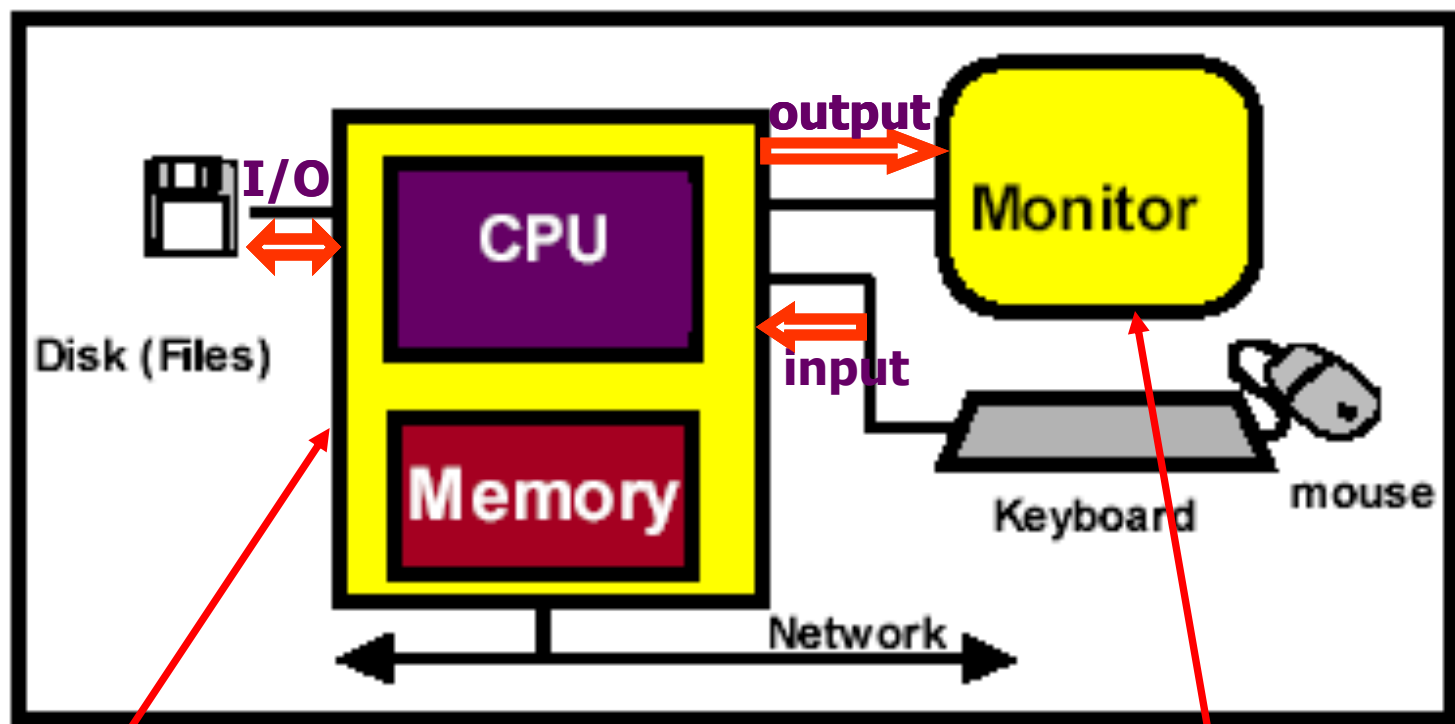
星期六，胜利大逃亡；

星期天，快乐的单身汉

请编写一个程序，帮小虎计算本月任意一天是星期几？

1. “发明”一个神奇数字 M ；
2. 对于本月任意一天 X ，计算 $(X+M) \% 7$ ；
3. 结果是几就是星期几。

1.3.3 输入与输出



核心部件

I/O: 相对于内存

I/O设备

字符界面 VS. 图形界面



C++的输入输出

在C++ 中引入术语 **stream**（流），指的是来自设备或传给设备的一个数据流。

在输入操作中，字节从输入设备流向内存

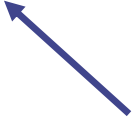
在输出操作中，字节从内存流向输出设备

头文件 **iostream**中定义了输入流 **cin**和输出流**cout**对象。

用cout进行输出

cout表示输出流对象，与它相关联的设备是显示器。**cout** 必须和输出运算符 **<<** 一起使用，表示将其后面的数据插入到输出流中去。

cout = c + out;



char

cout 举例

```
cout << "医生问病人是怎么骨折的。"<< endl;  
cout << "病人说：昨天我走在马路上，";  
cout << "觉得鞋里有沙子，就扶着电线杆抖鞋。\\n";  
cout << "有个人经过那里，以为我触电了，便抄  
起木棍给了我两棍子!" << endl;
```

```
#include <iostream>
using namespace std;
int main()
{
    int    i = 1;
    double f = 3.14;
    char  c = 'A', str[ ] = "hello, world\n";
    cout << i;
    cout << f;
    cout << c;
    cout << str;
    cout << i << f << c << str << endl;
}
```

用cin进行输入

cin表示输入流对象，与它相关联的输入输出设备是键盘。当我们从键盘输入字符串时，形成了输入流（数据流），用提取操作符 **>>** 将数据流存储到一个事先定义好的变量中。

cin = c + in;

cin举例

```
int    i;  
float  f;  
char  c, str[20];  
cin >> i >> f >> c >> str;
```

以空白字符（空格、制表符、换行、回车）
作为输入数据之间的间隔。

输入一个字符可以使用： `cin.get(c);`

输入一个字符串可用： `cin.getline(str, 20);`

1.3.4 GPA程序

- 让用户输入成绩分别为A、B、C的课程数
 - 计算总分数（分子）
 - 计算总的课程数（分母）
 - 计算GPA
 - 在屏幕上显示计算结果
1. 变量的定义与使用
 2. 算术运算
 3. 输入和输出

```

#include <iostream>    /* 编译预处理, 文件包含 */
using namespace std;

int main( )           /* 主函数 */
{
    int nA, nB, nC;    /* 成绩为A、B、C的课程数*/
    double totalCourses, totalPoints, gpa;

    cout << "你得了多少个A? " << endl;
    cin >> nA;        //输入成绩为A的课程数
    cout << "你得了多少个B? " << endl;
    cin >> nB;
    cout << "你得了多少个C? " << endl;
    cin >> nC;

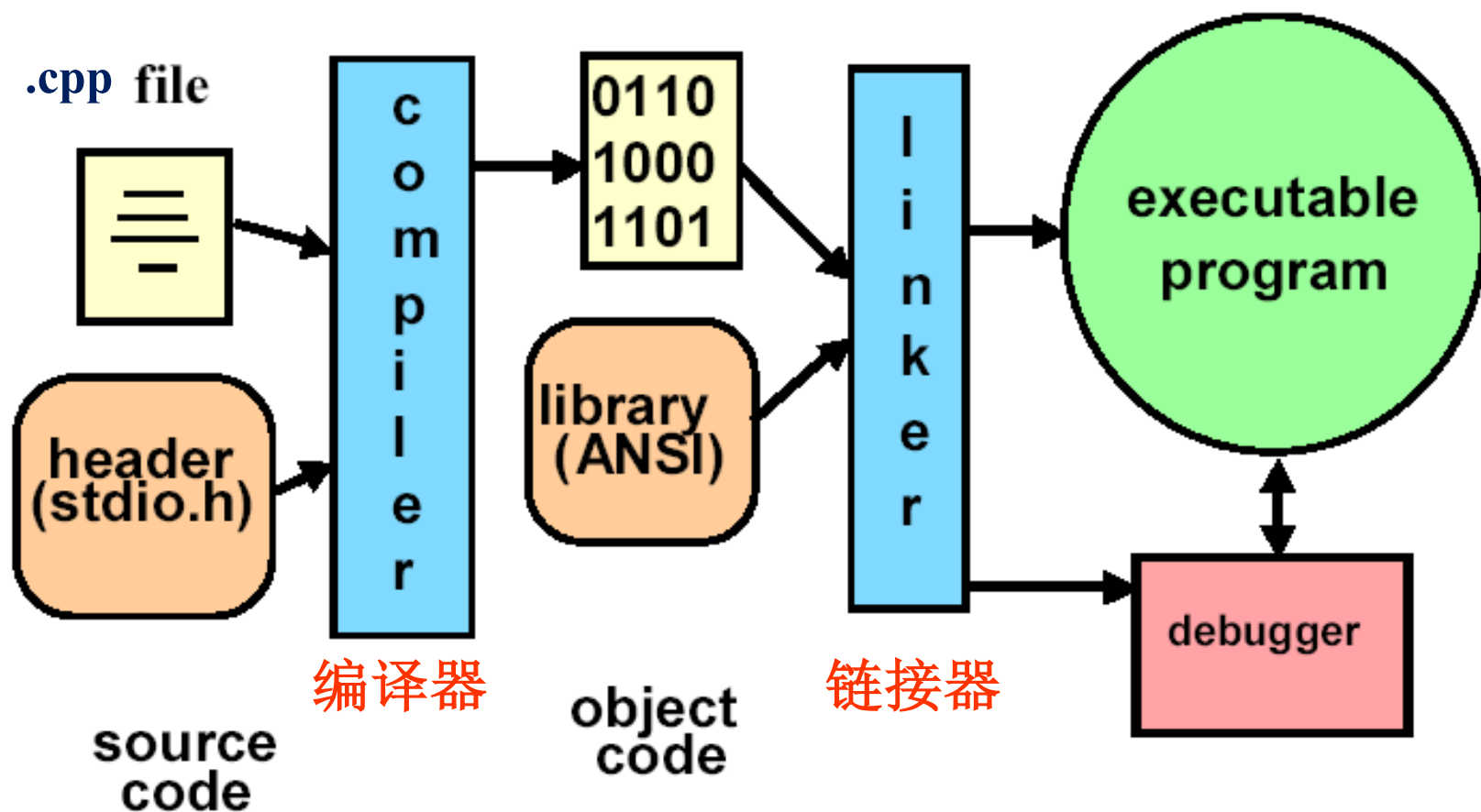
    totalPoints = nA * 4 + nB * 3 + nC * 2;
    totalCourses = nA + nB + nC;
    gpa = totalPoints / totalCourses;
    cout << "你的GPA是: " << gpa << endl;
    return (0);
}

```

源程序与可执行程序

- 用C++语言编写的程序称为“源程序”（source code）；
- 计算机的处理器CPU只能看懂用它自己的机器语言来编写的“可执行程序”（executable program）；
- 如何让CPU来执行用C++语言编写的程序？

从源程序到可执行程序的转换



一次运行结果

你得了多少个A? 1
你得了多少个B? 1
你得了多少个C? 1
你的GPA是: 3.00

1.4 R进制

在程序设计中，会接触到二进制、八进制和十六进制，其共同之处都是进位计数制。



如果某种数制只采用**R**个基本符号，则称为基**R**数制（**R**进制），**R**称为数制的“基数”，而数制中每一固定位置所对应的单位值称为“权”。

进位计数制的编码符合“逢**R**进一，借一当**R**”的规则，各个位的权是以**R**为底的幂，一个数可按照权展开成多项式。例如一个十进制数**(2564)₁₀**可按权展开为：

$$2564 = 2 \times 10^3 + 5 \times 10^2 + 6 \times 10^1 + 4 \times 10^0$$

几种常用的进位数制

二进制	R=2	基本符号	0, 1
八进制	R=8	基本符号	0, 1, 2, 3, 4, 5, 6, 7
十进制	R=10	基本符号	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
十六进制	R=16	基本符号	0 – 9, A, B, C, D, E, F

其中，十六进制的符号A~F分别对应于十进制的10~15。

R进制转换为十进制

基数为R的数，只要将其各位数字与相应的权相乘，其积相加，和数就是相应的十进制数。

$$\begin{aligned}\text{例: } 11001010_2 \\ &= 1 \times 2^1 + 1 \times 2^3 + 1 \times 2^6 + 1 \times 2^7 \\ &= 202\end{aligned}$$

$$\begin{aligned}\text{例: } 3407_8 \\ &= 7 \times 8^0 + 0 \times 8^1 + 4 \times 8^2 + 3 \times 8^3 = 1799\end{aligned}$$

$$\begin{aligned}\text{例: } 3C_{16} \\ &= C \times 16^0 + 3 \times 16^1 = 60\end{aligned}$$

清华大学计算机系第31届学生节



12月22日 晚6:30 大礼堂 敬请期待

@清华计算机系校友会

十进制转换为R进制

用该十进制数连续地除以R，得到的余数即为R系统的各位系数。此方法称为**除R取余法**。

例如：将 59_{10} 转换为二进制数：

2		59		余数	
2		29	1	低位
2		14	1	
2		7	0	
2		3	1	
2		1	1	
		0	1	高位

所以：

$$59_{10} = 111011_2$$

二进制与十六进制的转换

这两种进制的权之间有内在联系，即 $2^4=16$ ，因此它们之间的转换比较容易，即每位十六进制数相当于四位二进制数。

例如：将 1011010_2 转换成十六进制数：

$$\begin{array}{ccccccc} 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \end{array} = \begin{array}{cc} 0101 & 1010 \\ & \\ & \\ & \\ & \\ & \\ & \\ & \end{array} = 5A_{16}$$

5 A

将十六进制数F728转换为二进制数：

$$\begin{array}{ccccccc} F & 7 & 2 & 8 & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \end{array} = \begin{array}{cccc} 1111 & 0111 & 0010 & 1000 \end{array} = 1111011100101000_2$$

二进制数	八进制数	十六进制数	十进制数的值
0000	00	0	0
0001	01	1	1
0010	02	2	2
0011	03	3	3
0100	04	4	4
0101	05	5	5
0110	06	6	6
0111	07	7	7
1000	10	8	8
1001	11	9	9
1010	12	A	10
1011	13	B	11
1100	14	C	12
1101	15	D	13
1110	16	E	14
1111	17	F	15

无符号二进制数的取值范围

- 假设我们有4个二进制位：
 - 它能够表示的最小数为 $0000_2 = 0_{10}$;
 - 它能够表示的最大数为 $1111_2 = 15_{10}$;
- 假设我们有n个二进制位：
 - 它能够表示的最小数为 **0**;
 - 它能够表示的最大数为 **$2^n - 1$** ;

例如:

$n = 8$, 取值范围: $0 \sim 255(2^8 - 1)$;

$n = 16$, 取值范围: $0 \sim 65535(2^{16} - 1)$;

$n = 32$, 取值范围: $0 \sim 2^{32} - 1$ 。

有符号的情形

- 假设我们有4个二进制位：XXXX

补码	十进制	补码	十进制
0000	0	1000	-8
0001	1	1001	-7
0010	2	1010	-6
0011	3	1011	-5
0100	4	1100	-4
0101	5	1101	-3
0110	6	1110	-2
0111	7	1111	-1

- 数值是以补码的形式来表示；
- 正数的补码即为原码；
- 负数的补码等于其绝对值的二进制形式按位取反再加1；
- 最高位即符号位。

有符号二进制数的取值范围

- 假设我们有4个二进制位：
 - 它能够表示的最小数为 -8;
 - 它能够表示的最大数为 7;
- 假设我们有n个二进制位：
 - 它能够表示的最小数为 -2^{n-1} ;
 - 它能够表示的最大数为 $2^{n-1}-1$;

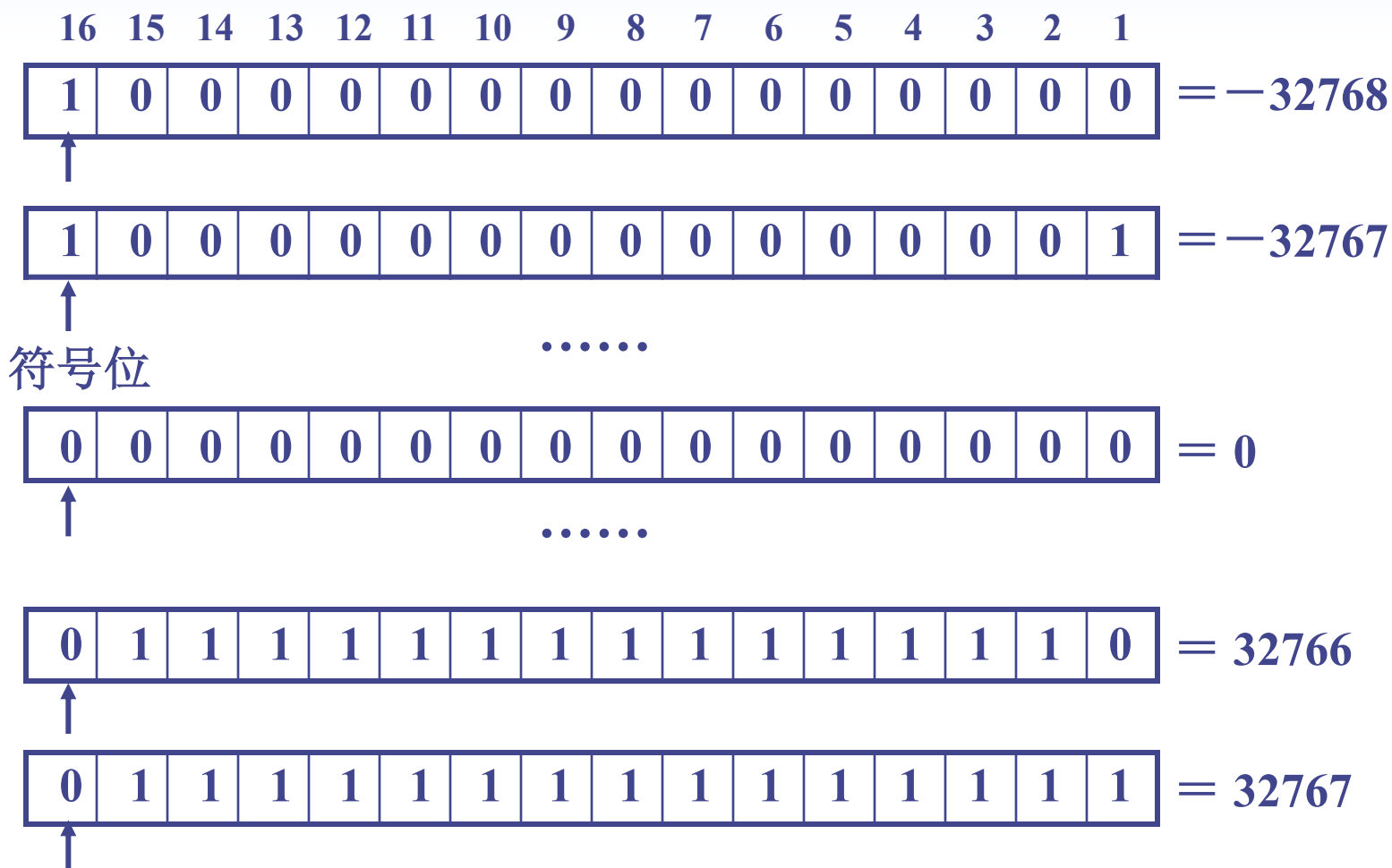
例如:

$n = 8$, 取值范围: $-128(-2^7) \sim 127(2^7 - 1)$;

$n=16$, 取值范围: $-2^{15} \sim 2^{15} - 1$;

$n=32$, 取值范围: $-2^{31} \sim 2^{31} - 1$ 。

有符号短整型数的取值范围



short x = 32767; x = x + 1;

国王的牛奶

很久以前，有一个国王，在他的仓库里存放着1000桶牛奶。

一天，有一个刺客在其中一桶牛奶中放了一种无色无味的毒药，这种毒药不论稀释多少倍都有毒，且没有任何东西能检测出此毒，只能用老鼠来测试，而且老鼠喝了要过一周才死亡。但一周之后国王就要用这批牛奶了，而且要用999桶。因此只有一周的时间来找出哪一桶牛奶有毒。

问题：至少需要用多少只老鼠做测试？

0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7



本讲小结

- ◆ 计算机程序的概念
- ◆ 数据类型
- ◆ 变量
- ◆ 基本输入输出
- ◆ 数的进制和表示