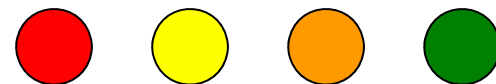




程序设计基础 (C语言)

郎大鹏



第1章 程序设计基础

- 1.1 程序设计概念
- 1.2 算法的概念及其表示
- 1.3 程序开发过程
- 1.4 C语言程序结构及实例



1.1 程序设计概念

什么是计算机？

- 计算机是用来延伸人的能力的工具，需要人来驾驭
- 我们的职责是让非计算机专业的人更容易驾驭它
- 完成这一目标的主要手段之一就是



“编程 (Programming)”

1.1 程序设计概念

什么是程序设计？

- 解决特定问题是程序设计的目的，程序设计的最终目的就是用程序来控制计算机来为人们解决特定的问题。

程序设计一般包括以下几个步骤：

- (1) 分析待解决的问题并建立相应的数学模型；
- (2) 确定数据结构和算法。根据建立的数学模型，确定存放数据的数据结构，针对所确定的数据结构选择合适的算法；
- (3) 编程；
- (4) 调试程序；
- (5) 整理并写出文档资料。



1.1 程序设计概念

什么是计算机程序？

- 人们要利用计算机完成各种预定的工作，就必须把完成该项工作所需要的步骤编写成计算机可以执行的指令，**程序**即是为实现特定目标或解决特定问题而用计算机语言编写的指令序列的集合。



1.1 程序设计概念

什么是计算机程序？

- 一个程序应该包括两方面的内容：

对数据的描述，指定数据类型和组织形式，即是数据结构；

对操作步骤的描述，对操作步骤的描述即是算法。

正如著名计算机科学家沃思（Nikiklaus Wirth）提出的一个公式：

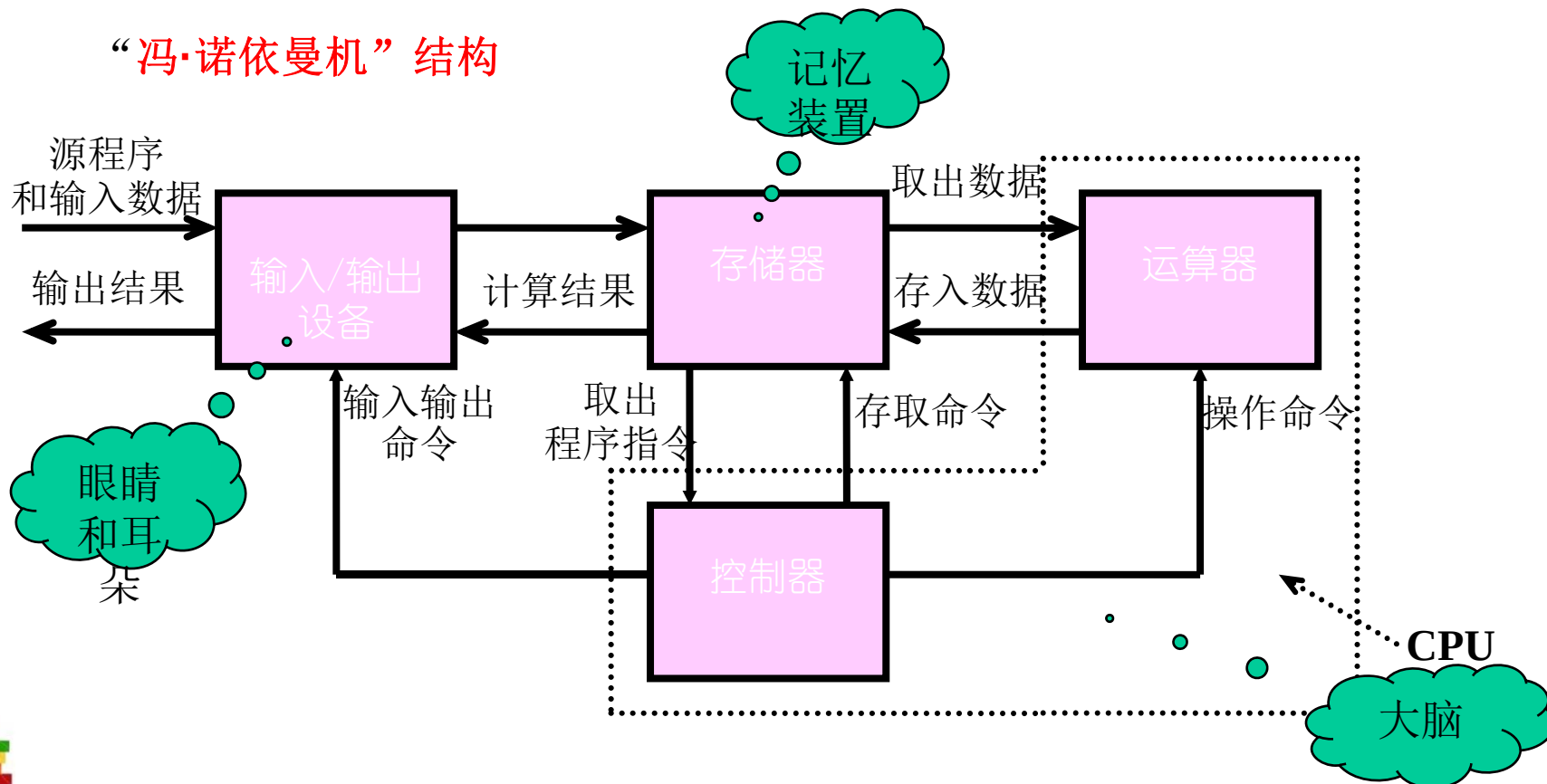
数据结构 + 算法 = 程序



1.1 程序设计概念

计算机基本工作过程

“冯·诺依曼机”结构

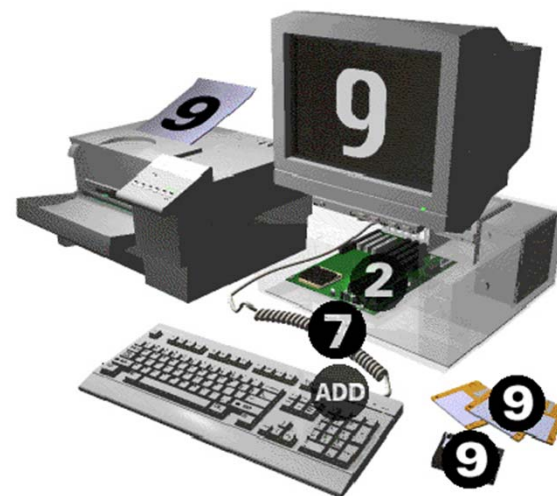


2013年2月1日

1.1 程序设计概念

程序运行过程

- 软件的运行
 - 计算机把机器代码读入到内存 (Memory)
 - 由CPU运行这些代码
 - 读取输入 (Input)
 - 产生输出 (Output)
 - 完成程序员预定的功能
- 计算机专业后续课程
 - 计算机原理
 - 编译原理
 - 操作系统



1.1 程序设计概念

- 程序设计语言的发展

- 机器语言编写的1+1程序

```
10111000
00000001
00000000
00000101
00000001
00000000
```

- 汇编语言（Assemble Language）编写的1+1程序

```
MOV AX, 1
ADD AX, 1
```



1.1 程序设计概念

- 程序设计语言的发展

- BASIC语言编写的1+1程序

```
PRINT 1+1
```

- C语言编写的1+1程序

```
#include <stdio.h>
main()
{
    printf("%d\n", 1+1);
}
```



1.1 程序设计概念

- 程序设计思想

1. 结构化程序设计思想—模块化

结构化程序设计的概念是E.W.Dijkstra在20世纪60年代末提出的，其实质是控制编程中的复杂性。它的主要观点是采用自顶向下、逐步求精的程序设计方法

具体来说，采用以下方法确保得到结构化的程序：

- (1) 自顶向下
- (2) 逐步求精
- (3) 模块化设计
- (4) 结构化编码



1.1 程序设计概念

- 程序设计思想

1. 面向对象程序设计思想

面向对象程序设计是面向过程设计方法的继承和发展，是基于一种自然、朴素的思想来模拟人类思维习惯的程序设计方法。它具有抽象、封装、继承和多态性等特征。客观世界是由各种各样的对象组成的，每个对象都有自己的内部状态和运动规律，不同对象间的相互联系和相互作用构成了各种不同的系统，进而构成整个客观世界。

C++语言是以C语言为基础的一种面向对象的语言，有着极为广泛的应用。



1.2 算法的概念及表示

- 什么是算法

1. 广义上来说，任何解决问题的过程都是由一定的步骤组成的，这些解决问题确定的方法和有限的步骤都可以称作算法。

2. 常计算机算法分为两大类：数值运算算法和非数值运算算法。

数值运算是指求数值解，例如对微分方程求解、对函数的定积分求解等，都属于数值运算的范围。

非数值运算包括的范围非常广，如人事管理、数据处理、资料检索等。



1.2 算法的概念及表示

- 算法的特征

- (1) 确定性
- (2) 有穷性
- (3) 有效性
- (4) 输入
- (5) 输出



1.2 算法的概念及表示

- 算法的复杂性

时间复杂性

算法的时间复杂性越高，则算法的执行时间越长；时间复杂性越低，则执行时间越短。

空间复杂性

算法的空间复杂性越高，算法所需的存储空间越多；空间复杂性越低，则所需的存储空间越少。

算法的复杂性分析中，对时间复杂性的分析考虑得更多。



1.2 算法的概念及表示

- 常用算法简介

枚举法

枚举法又称为穷举法，将问题的所有可能的答案一一列举，然后根据条件判断此答案是否合适，合适就为本题的一个答案，不合适就丢弃。若全部可能答案均不符合条件，则说明该题无答案。

迭代法

迭代法是一种不断用变量的旧值递推新值的过程，是一种数值近似求解的方法，迭代算法是用计算机解决问题的一种常见的方法。



1.2 算法的概念及表示

- 常用算法简介

递归法和递推法

枚举法递推法是一种比较简单的算法，通过已知条件，利用递推公式，由简到繁逐次迭代，直至得到结果的算法。递推法的关键是找到进行递推的通项公式。

递归法也是利用递推公式，但不同的是，它是由繁化简，用简单的问题和已知的操作运算来解决复杂的问题。

分治法

分治法就是把一个复杂的问题分解成两个或更多的相同或相似的子问题，再把子问题分成更加小的子问题，直到最后子问题可以简单直接的求解，最后再把子问题的解合并，就组合成原问题的解。



1.2 算法的概念及表示

- 算法的表示

用自然语言表示算法

例1.3 输入3个数，输出最小的数。

首先定义3个变量 x 、 y 、 z ，将3个数依次输入到 x 、 y 、 z 中，再定义一个变量 MIN ，存放最小数。算法步骤如下：

- (1) 输入 x ， y ， z 。
- (2) 若 $x < y$ ，则 $x \rightarrow MIN$ ，否则 $y \rightarrow MIN$ 。
- (3) 若 $z < MIN$ ，则 $z \rightarrow MIN$ 。



1.2 算法的概念及表示

用伪代码表示算法

例1.4 输入两个数 x 、 y ，输出最大的数。

开始

输入 x 、 y

if $x > y$

 输出 x

else

 输出 y

结束



1.2 算法的概念及表示

用流程图表示算法；

规则：从上到下，从左到右。

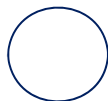
符号及其意义：

流程线 

起止框

处理框

连接点



判断框

I/O框



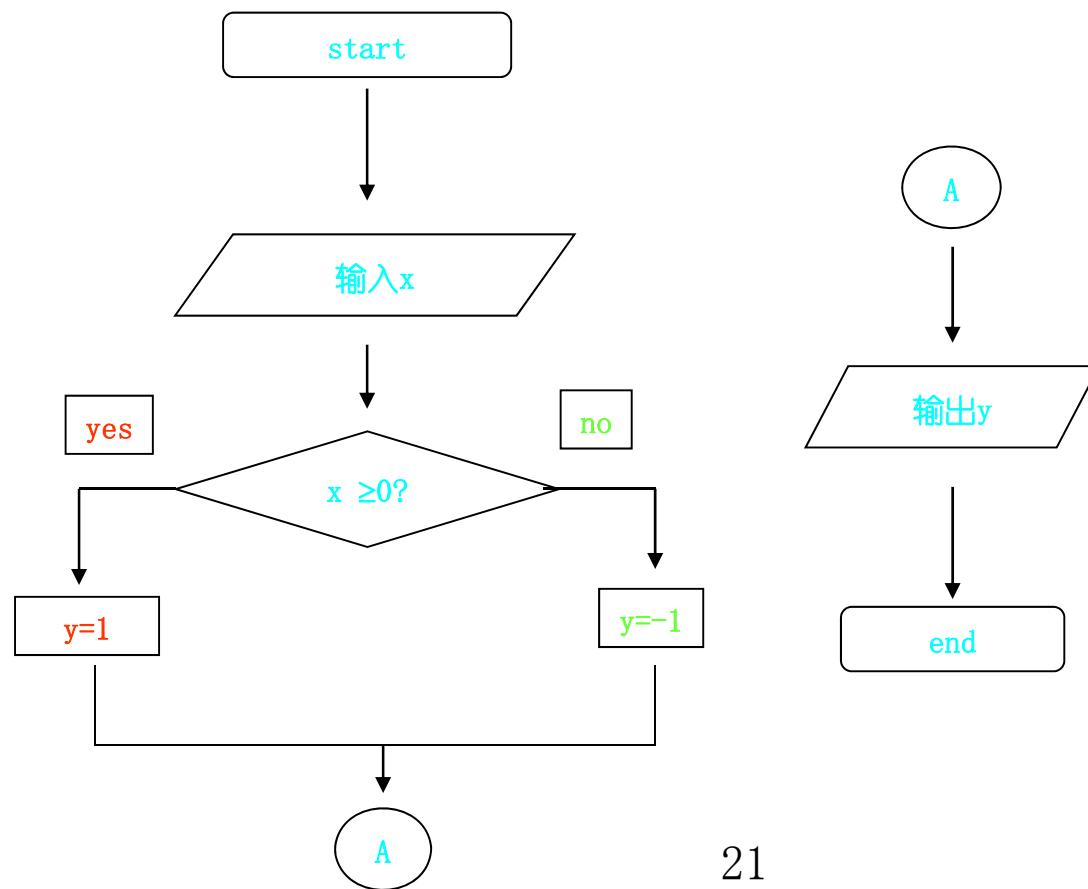
1.2 算法的概念及表示

- 算法的表示

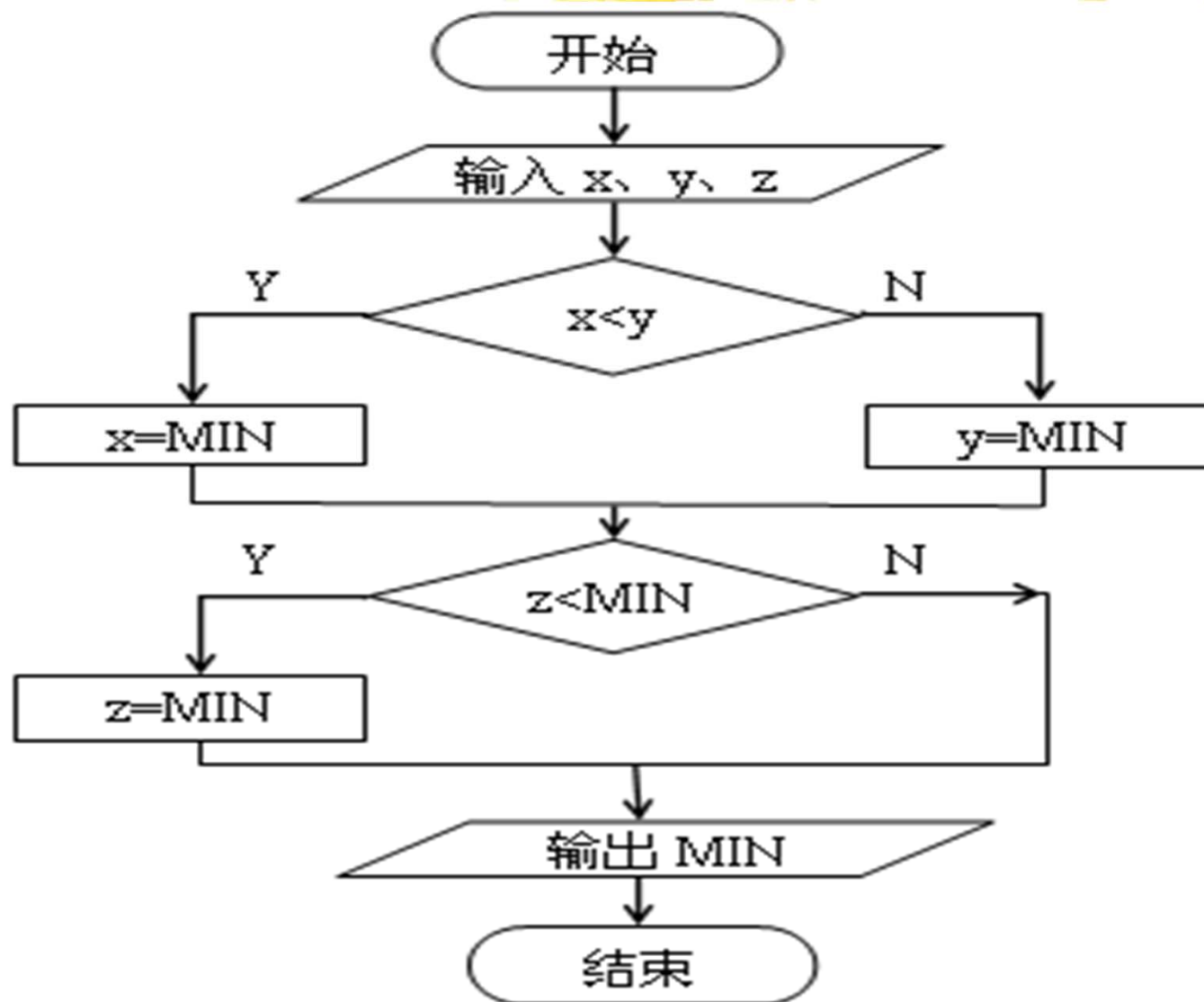
流程图表示举例

用框图描述函数

$$y = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases}$$



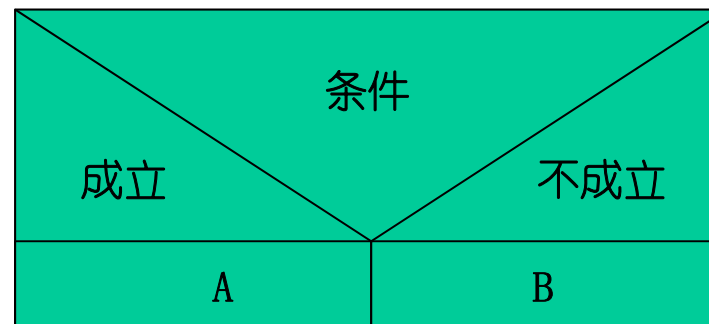
1.2 算法的概念及表示



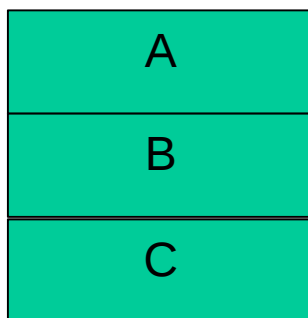
1.2 算法的概念及表示

- 算法的表示

N-S流程图表示算法

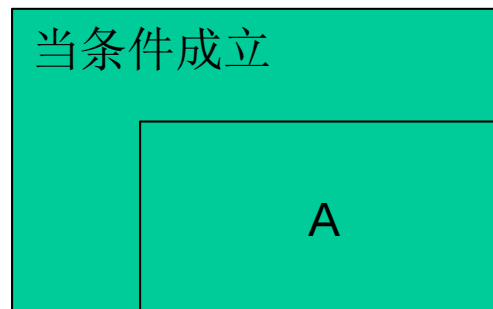


分支结构

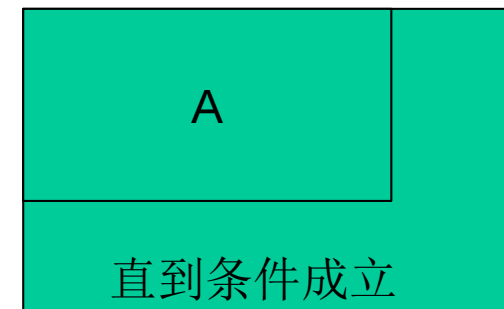


顺序结构

当型循环

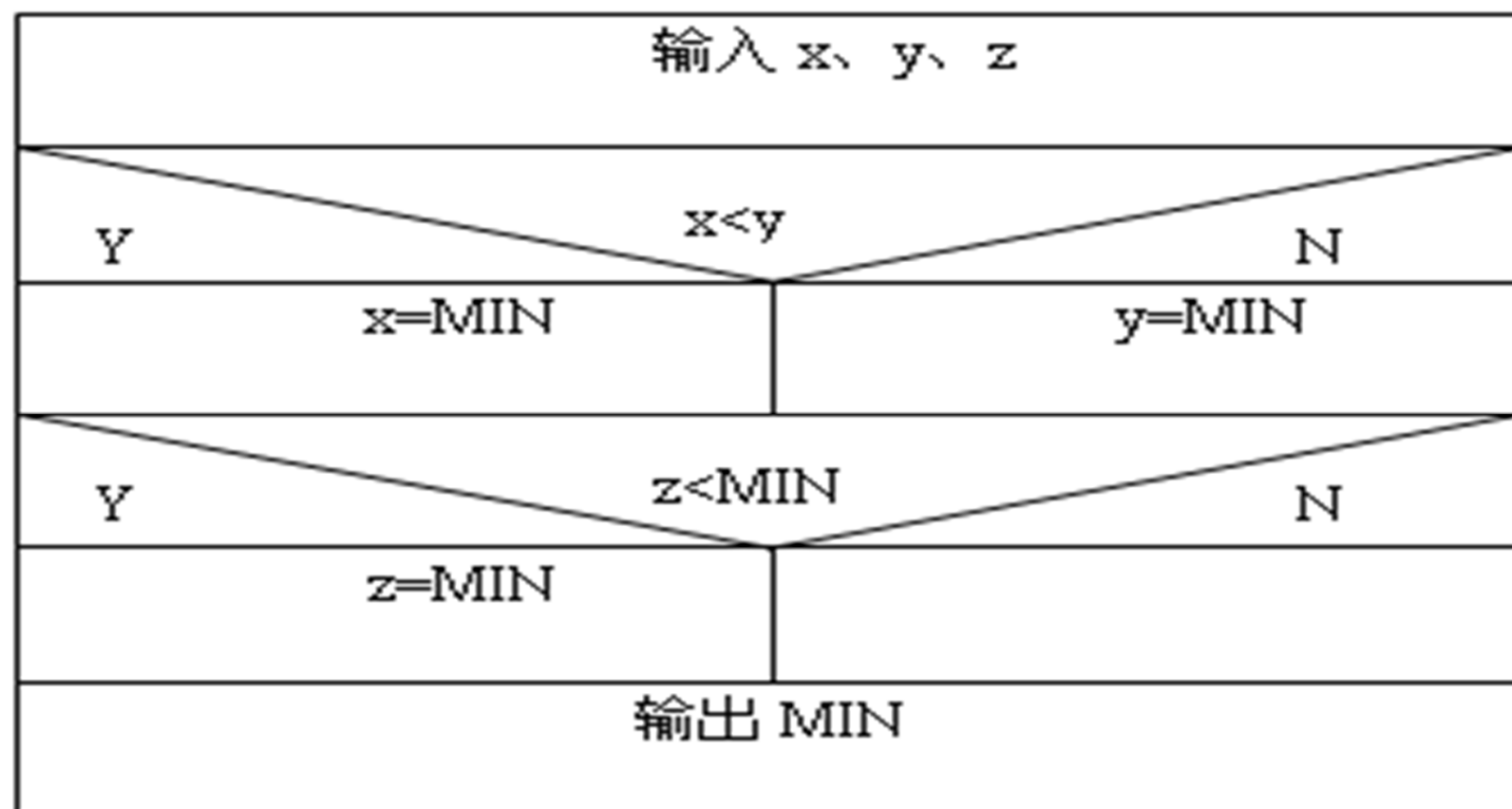


直到型循环



循环结构（条件在前/条件在后）





1.3 程序的开发过程

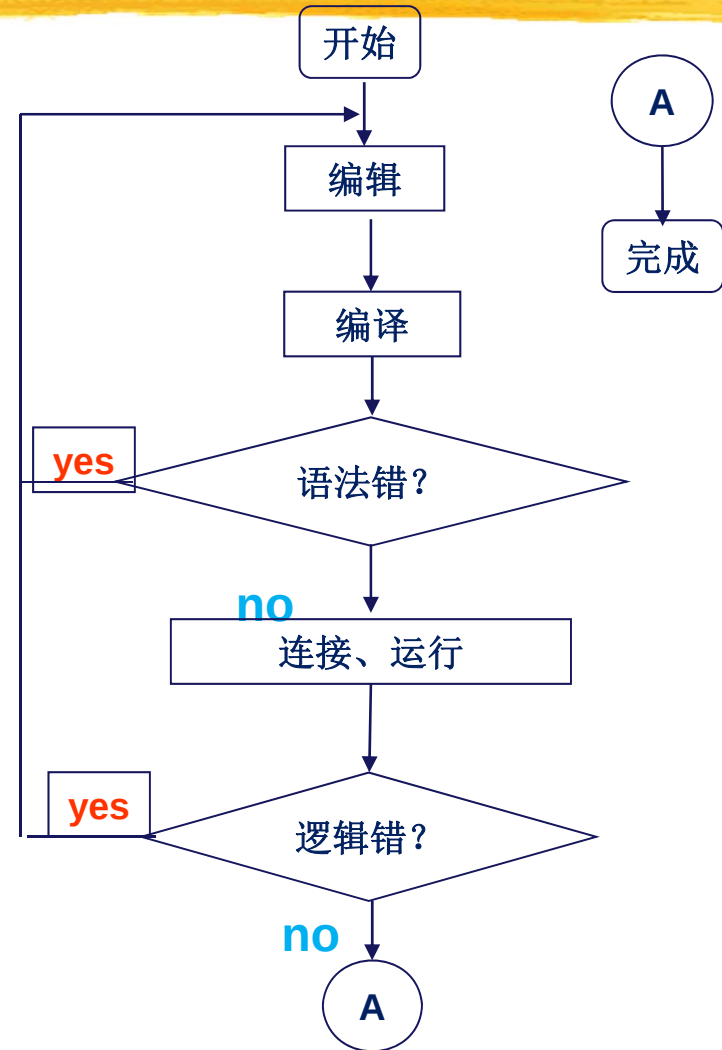
- (1)系统描述
- (2)需求分析 描述系统功能
- (3)结构分析 分析设计，算法描述
- (4)详细设计 编码
- (5)代码调试
- (6)单元测试 调试
- (7)系统测试
- (8)运行维护



1.3 程序的开发过程

上机过程

- 编辑 (Edit)
 - 扩展名为.c的文件
- 编译 (Compile)
 - 扩展名为.obj的文件
- 链接 (Link)
 - 形成最后的可执行文件
 - 扩展名为.exe的文件
- 运行 (Run)



1.4 C语言程序结构及实例

- C语言是当今世界上最为流行的计算机高级语言之一，它是1972年由美国的Dennis Ritchie设计发明的，并首次在UNIX操作系统的DEC PDP-11 计算机上使用。

C语言的设计者
Dennis M. Ritchie



和Unix的设计者Ken Thompson
接受美国国家技术勋章



1.4 C语言程序结构及实例

- 目前最流行的**C**语言版本有以下几种：

- **Microsoft C**或称**MSC**

- **Borland Turbo C**或称**Turbo C**

- **AT&T C**

这些**C**语言版本不仅实现了**ANSI C**标准，而且在此基础上各自作了一些扩充，使之更加方便、完美。



1.4 C语言程序结构及实例

例1 在DOS屏幕上输出This is a C program.

程序如下:

```
void main()
{
    printf("This is a C program."); //函数体
}
```

该程序的功能是:

执行该程序, 将在屏幕上输出一行信息:

This is a C program .



1.4 C语言程序结构及实例

例2 输入两个数及其运算关系，输出运算结果。

```
#include <stdio.h>
void main(void)
{
    char ch;
    int a, b;
    int c;
    scanf("%d%c%d",&a,&ch,&b);
    c=fun(a,ch,b);
    printf("a%cb=%d",ch,c);
}
```

```
int fun(int x,char op,int y)
{
    int z;
    switch(op)
    {
        case '+': z=x+y; break;
        case '-': z=x-y; break;
        case '*': z=x*y; break;
        case '/': z=x/y; break;
    }
    return z;
}
```

子
函
数

3.4 程序是由函数组成的。每个函数是由语句组成的。其中主函数是唯一的，主函数名位
main。但不函数到结构。无也可以有多个。



1.4 C语言程序结构及实例

- 标识符和关键字

- 1. C语言的字符集

- (1)字母: A、a、B、b...、Z、z 52个
 - (2)数字符号: 0、1...、9 10个
 - (3)特殊符号: + - * /... 20个

- 2. 关键字 (又称保留字)

系统占用词 (共32个)

auto、break、case、char、const、continue、default、do、double、else、enum、extern、float、for、goto、if、int、long、register、return、short、signed、sizeof、static、struct、switch、typedef、union、unsigned、void、volatile、while

建议不要使用:

define、undef、include、ifdef、ifndef、endif、line、error、elif、pragma



1.4 C语言程序结构及实例

- 标识符和关键字

3. 标识符

组成规则：

- (1)字母和数字以及 下划线_ 组成。
- (2)以字母和 下划线_ 开头。
- (3)长度不超过32个字符。
- (4)不能使用保留字。

注意：系统区分大小写字母。

aBc

Abc

_iArea

2Ab

_fLoad

int

ERROR!

iStatus

faS



作业

- 1.1 简述C语言的特点。
- 1.2 有三个数A、B、C，设计算法，求三个数中最大的数并输出。（可用自然语言或流程图表示）。
- 1.3 设计算法，求N个整数的平均值。
- 1.4 简述结构化程序设计的基本思想。
- 1.5 上机运行本章例1.6、例1.7。
- 1.6 请参照本章例题，编写一个C程序，输出以下信息：

This is my first C program!

