

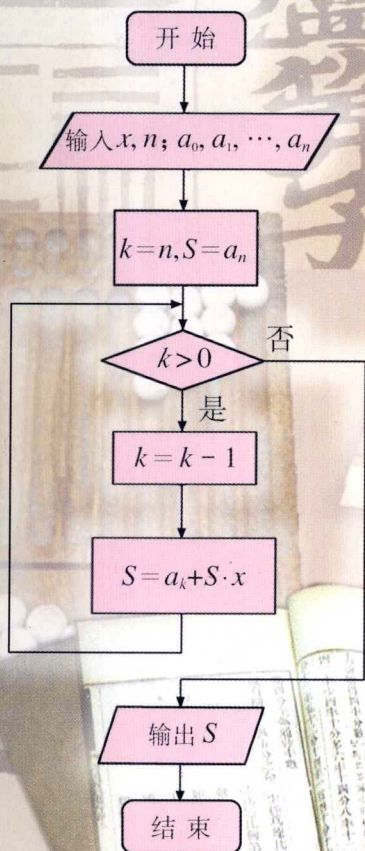
经全国中小学教材审定委员会
2004年初审通过

普通高中课程标准实验教科书

数学 ③

必修

人民教育出版社 课程教材研究所 编著
中学数学教材实验研究组



秦九韶算法框图

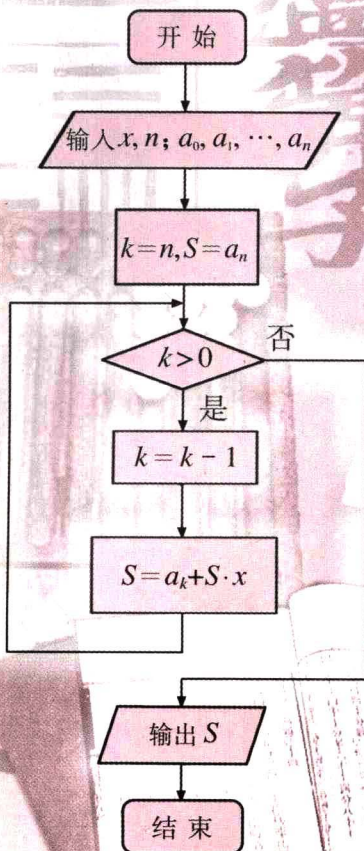
普通高中课程标准实验教科书

数学

3

必修

人民教育出版社 课程教材研究所 编著
中学数学教材实验研究组



秦九韶算法框图



人民教育出版社

版

主 编 高存明

本册主编 高尚华

编 者 陈 研 高尚华 王旭刚
陈亦飞 王殿军 黄 铎

责任编辑 王旭刚

美术编辑 张 蓓 王 喆

绘 图 王 鑫

封面设计 林荣桓

普通高中课程标准实验教科书

数学 3

必修

B 版

人民教育出版社 课程教材研究所 编著
中学数学教材实验研究组

*

人民教育出版社出版发行

(北京沙滩后街 55 号 邮编: 100009)

网址: <http://www.pep.com.cn>

人民教育出版社印刷厂印装 全国新华书店经销

*

开本: 890 毫米×1 240 毫米 1/16 印张: 8.75 字数: 174 000

2004 年 5 月第 1 版 2004 年 12 月第 3 次印刷

ISBN 7-107-17712-5 定价: 9.55 元
G·10801 (课)

如发现印、装质量问题, 影响阅读, 请与出版社联系调换。

(联系地址: 北京市方庄小区芳城园三区 13 号楼 邮编: 100078)

本册导引

欢迎同学们进入数学3的学习。本模块将学习三部分内容：算法、统计和概率。

“算法”在数学1就已进入你的视野，你已看到用数学语言描述过的算法。你可能会觉得算法比较机械、刻板甚至有点枯燥。这一章我们将较系统地学习算法，通过这一章的学习，你一定会体会到学习算法的重要作用，而且你也一定能感受到学习算法的乐趣。算法一章主要是要求同学们掌握算法的基本思想，能写出或用程序框图画简单数学问题的算法。希望有条件使用计算机的同学能用程序设计语言编写程序，通过上机运算来体会算法思想的重要性和有效性。

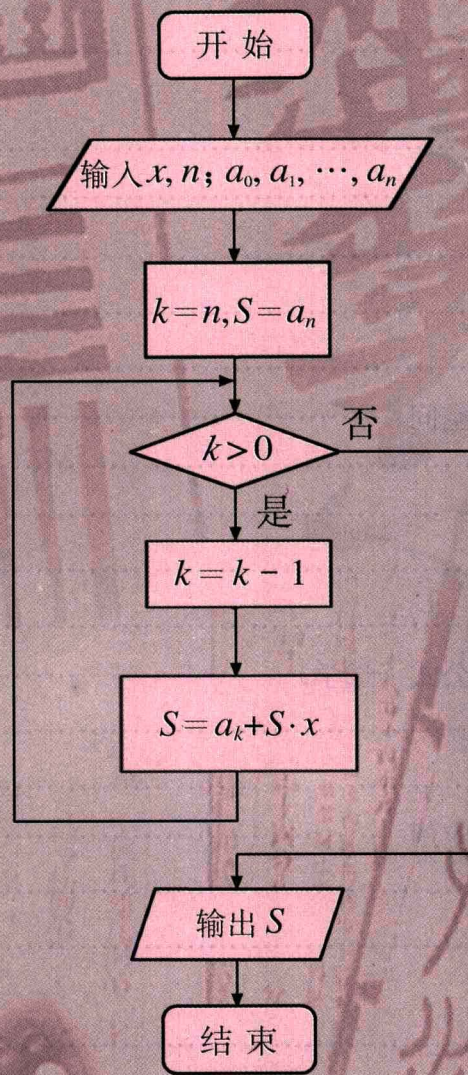
本模块的第二章和第三章内容是统计和概率。

在义务教育阶段，同学们初步接触到统计，获得了一些统计知识。在此基础上本模块介绍若干抽样方法，通过实例，学习用样本估计总体的思想，进一步体会统计思想与统计方法。收集、整理、分析数据是统计学的基本内容，通过收集大量的数据，从中提取有价值的信息，经过数据分析，对现象作出判断，以对“行为”作出合理的建议或决策。这就是统计学的任务。

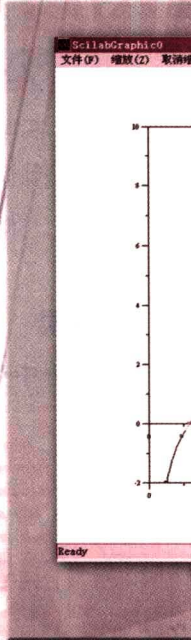
概率与统计都是研究随机现象的，它们有着深刻的内在联系。概率为人们认识随机现象和解决这类实际问题提供了有效的方法，同时为统计学的发展提供了理论基础。这一章将通过观察与研究，了解概率的基本性质和几种概率模型，从中体会、理解概率的意义和在现实生活中的重要作用。

在统计与概率的学习中，教材力求将算法融入统计与概率的学习之中，一方面让同学们知道，如何应用统计与概率思想设计好的算法，另一方面可以进一步体会算法的精要，帮助我们解决实际的统计与概率问题。

算法、统计、概率的基础知识，已经成为未来公民的必备常识，它们的共同特点是富于实践性。同学们可以通过实践领会它们的精要，反过来，可以应用它们的思想与方法去解决实际问题。虽然同学们对它们的内容比较生疏，但它们的可实践性将帮助大家度过难关。我们相信，同学们会怀着强烈的求知欲望和浓厚的兴趣投入本模块的学习活动。



秦九韶算法框图



目 录

第一章 算法初步	1
1.1 算法与程序框图	3
◆ 1.1.1 算法的概念	3
◆ 1.1.2 程序框图	8
◆ 1.1.3 算法的三种基本逻辑结构和框图表示	11
1.2 基本算法语句	20
◆ 1.2.1 赋值、输入和输出语句	21
◆ 1.2.2 条件语句	25
◆ 1.2.3 循环语句	27
1.3 中国古代数学中的算法案例	33
本章小结	40
阅读与欣赏	
我国古代数学家秦九韶	42
东方数学的使命	42
附录 参考程序	44
第二章 统计	47
2.1 随机抽样	49
◆ 2.1.1 简单随机抽样	50
◆ 2.1.2 系统抽样	53
◆ 2.1.3 分层抽样	54
◆ 2.1.4 数据的收集	55
2.2 用样本估计总体	60
◆ 2.2.1 用样本的频率分布估计总体的分布	60
◆ 2.2.2 用样本的数字特征估计总体的数字特征	68
2.3 变量的相关性	76
◆ 2.3.1 变量间的相关关系	76
◆ 2.3.2 两个变量的线性相关	78

实习作业	86
本章小结	89
阅读与欣赏	
蚂蚁和大象谁的力气更大	92
附录 随机数表	93
第三章 概率	95
3.1 事件与概率	97
◆ 3.1.1 随机现象	97
◆ 3.1.2 事件与基本事件空间	99
◆ 3.1.3 频率与概率	102
◆ 3.1.4 概率的加法公式	105
3.2 古典概型	110
◆ 3.2.1 古典概型	110
◆ 3.2.2 概率的一般加法公式 (选学)	115
3.3 随机数的含义与应用	118
◆ 3.3.1 几何概型	118
◆ 3.3.2 随机数的含义与应用	119
3.4 概率的应用	125
本章小结	129
阅读与欣赏	
概率论的起源	131
附录	
部分中英文词汇对照表	132

第一章 算法初步

SciPad - F:/SCI/LAB/12月16日/2.SCI
File Edit Search Windows Options Load into Scilab Help

```
if d1<=0 then  
    x_message('初始数据错误!', '要求两端点函数值相异', ['确定']);  
else  
    d1=0 then  
        x_message('方程的近似解是', string(x1), ['确定']);  
    else  
        d2=0 then  
            x_message('方程的近似解是', string(x2), ['确定']);  
        else  
            while (abs(x2-x1)>= e & flag=0)  
                m=(x1+x2)/2;  
                x=m;  
                dm=evatr(funs);  
                w=1;  
                d1=evatr(funs);  
                w=2;  
                dm=evatr(funs);  
                if (abs(x2-x1)>0.2  
                    & pause(2000000);  
                    color2d@pmod(m, (colorid=5, 32);  
                    w=1; , colorid);  
                    plot2d(w, dm, -6);  
                end  
                if dm=0  
                    flag=1;  
                end  
                if d1<=0  
                    x2=m;  
                else  
                    x1=m;  
                end  
            end  
            x_message('方程的近似解是', string(m), ['确定']);  
        end  
    end  
end
```

SciPad - D:/二分法.SCI
File Edit Search Windows Options Load into Scilab Help

```
title='二分法程序';  
twe='请输入函数, 例如 log(x)-2/x';  
twe1='请输入给定区间的左端点, 例如 1';  
twe2='请输入给定区间的右端点, 例如 30';  
twe3='请输入误差值e, 例如 0.0001';  
rute=x_mdialog(title, twe, 'log(x)-2/x', '1', '30', '0.0001');  
funs=rute(1);  
left_point=evatr(rute(2));  
right_point=evatr(rute(3));  
e=evatr(rute(4));  
flag=0;  
while (abs(right_point - left_point)>= e & flag=0)  
    mid_point=(left_point + right_point)/2;  
    x=mid_point; f_mid=evatr(funs);  
    if f_mid<0  
        left_point=x; f_left=evatr(funs);  
    else  
        right_point=x; f_right=evatr(funs);  
    end  
    if f_mid=0  
        flag=1;  
    end  
    if f_left*f_mid<0  
        right_point=mid_point;  
    else  
        left_point=mid_point;  
    end  
end  
x_message('函数'+funs+'的近似零点是', string(mid_point), ['确定']);
```

1.1

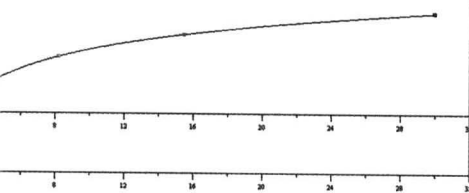
算法与程序框图

1.2

基本算法语句

1.3

中国古代数学中的算法案例



三维旋转 (R)

在数学1中，我们学习过求函数零点的二分法算法，在数学2的解析几何初步中，我们又把利用公式计算的几何问题进行分步求解。这些计算方法都有一个共同的特点，就是对一类问题（不是个别问题）都有效，计算可以一步一步地进行，每一步都能得到惟一的结果。通常我们把这一类问题的求解过程，叫做解决这一类问题的算法。算法一般是机械的，有时要进行大量重复的计算，它的优点是一种通法，只要按部就班地去做，总能算出结果。通常把算法过程称为“数学机械化”。数学机械化最大的优点，是它可以让计算机来完成。

在数学发展的历程中，寻求对一类问题的算法一直是数学发展的一个重要特点。在现代，“算法”已是数学及其应用科学中的重要组成部分，并成为计算科学的重要基础。随着现代信息技术的飞速发展，算法在科学技术和社会发展中发挥着越来越大的作用，并日益融入社会生活的许多方面。算法思想也逐渐成为每个现代人应具有的数字素养。

我们还要特别向同学们指出，我国古代数学发展的主导思想，就是构造“算法”解决实际问题。在这种思想的主导下，我国古代直到宋、元时期，数学的发展一直处于世界的领先地位。这一章的最后一节，我们要举例说明我国古代数学中蕴含的丰富的算法思想，以此引导同学们能沿着中国数学机械化的道路，学习数学，研究数学。

在用数学方法求解问题的过程中，提出问题、构造算法和使用计算工具，这三项工作始终紧密的联系在一起。从结绳计数、算筹到计算机的产生，计算工具的发展大大加快了研究算法的进程。现在我国高级中学大都拥有计算机，为学习算法提供了有力的工具。在这一章，我们将通过数学语言叙述、画程序框图和使用程序语言，学习基本算法语句的结构。

我们选用 Scilab 语言来实现算法。这种语言简单，容易学习，我们相信 Scilab 会成为同学们学习数值算法的好帮手。同学们可以使用 Scilab，将自己设计的算法在计算机上运行。你一定能获得成功的喜悦，做许多你用笔和纸不敢做的数学问题，为你今后在数学的学习中使用计算机技术打下良好的基础。



1.1

算法与程序框图

1.1.1 算法的概念

在前面的学习中，同学们已经接触到算法的概念，这一章我们专门来学习算法的知识。算法可以理解为由基本运算及规定的运算顺序所构成的完整的解题步骤，或者看成按照要求设计好的有限的确切的计算序列，并且这样的步骤或序列能够解决一类问题。

描述算法可以有不同的方式。例如，可以用自然语言和数学语言加以叙述，也可以借助形式语言（算法语言）给出精确的说明，也可以用框图直观地显示算法的全貌。

怎样才能设计出一个名副其实的算法呢？下面先从大家耳熟能详的鸡兔同笼问题谈起。

例 1 “一群小兔一群鸡，两群合到一群里，要数腿共 48，要数脑袋整 17，多少小兔多少鸡？”

求解这个问题的算术方法是：若没有小兔，则小鸡应为 17 只，总的腿数应为 $2 \times 17 = 34$ 。但现在有 48 条腿，造成腿的数目不够是由于假定小兔的数目为 0，每有一只小兔便会增加 2 条腿，故应该有

$$\frac{48 - 17 \times 2}{2} = 7$$

只小兔，相应地，小鸡则应有 10 只。

求解鸡兔同笼问题的上述方法简单直观，却又包含着深刻的算法思想。同学们都学过解二元一次方程组，现在让我们用解二元一次方程组的方法来重新求解鸡兔同笼问题：

设有 x 只小鸡， y 只小兔，则有

$$(I) \begin{cases} x+y=17 \\ 2x+4y=48 \end{cases}$$

将方程组(I)中的第一个方程的两边同乘以 -2 加到第二个方程中去，得到

$$(II) \begin{cases} x+y=17 \\ (4-2)y=48-17 \times 2 \end{cases}$$

解方程组(II)中的第二个方程，得 $y = \frac{48-17 \times 2}{4-2} = 7$ ，

将 y 代入第一个方程，得

$$x = 17 - y = 17 - 7 = 10$$

前面解决鸡兔同笼问题的算术方法，本质上是利用方程组(I)中的第一个方程来消去第二个方程中的未知数 x ，从而使该方程组(I)化为与其等价的方程组(II)，进而通过(II)的第二个方程确定 y ，再通过第一个方程确定 x 。这种求解方程组的方法称为高斯消去法。

这个方法显然可以用来解一般的二元一次方程组。现在我们给出用高斯消去法解一般的二元一次方程组

$$\begin{cases} a_{11}x_1 + a_{12}x_2 = b_1 & \text{①} \\ a_{21}x_1 + a_{22}x_2 = b_2 & \text{②} \end{cases}$$

的算法描述。

因为是二元一次方程组，所以方程组中 a_{11} ， a_{21} 不能同时为 0。

第一步：假定 $a_{11} \neq 0$ （如果 $a_{11} = 0$ ，可将第一个方程与第二个方程互换）， $\text{①} \times \left(-\frac{a_{21}}{a_{11}}\right) + \text{②}$ ，得到

$$\left(a_{22} - \frac{a_{21}a_{12}}{a_{11}}\right)x_2 = b_2 - \frac{a_{21}b_1}{a_{11}}$$

$$\text{即方程组可化为} \begin{cases} a_{11}x_1 + a_{12}x_2 = b_1 & \text{③} \\ (a_{11}a_{22} - a_{21}a_{12})x_2 = a_{11}b_2 - a_{21}b_1 & \text{④} \end{cases}$$

第二步：如果 $a_{11}a_{22} - a_{21}a_{12} \neq 0$ ，解方程④得到

$$x_2 = \frac{a_{11}b_2 - a_{21}b_1}{a_{11}a_{22} - a_{21}a_{12}} \quad \text{⑤}$$

第三步：将⑤代入③，整理得到

这里方程组的写法与以前学到的不同。方程组的未知数用 x_1 和 x_2 表示， a_{11} 表示方程组中的第一个方程第一个未知数 x_1 的系数， \dots ， a_{22} 表示第二个方程中第二个未知数 x_2 的系数， b_1 表示第一个方程的常数项， b_2 表示第二个方程的常数项。

$$x_1 = \frac{a_{22}b_1 - a_{12}b_2}{a_{11}a_{22} - a_{21}a_{12}} \quad \textcircled{6}$$

第四步：输出结果 x_1, x_2 。

如果 $a_{11}a_{22} - a_{21}a_{12} = 0$ ，则从④可以看出，方程组无解或有无穷多组解。

以后，我们在描述算法时，用英文 Step1, Step2, …，来表示第一步，第二步，…，也可以简写为：S1, S2, …。

从以上计算可以看出， $a_{11}a_{22} - a_{21}a_{12}$ 是一个很重要的值，它决定了方程组是否有惟一解。

上面得到的结果⑤、⑥，叫做求解二元一次方程组两个未知数的公式。利用这组公式，我们可写出解二元一次方程组的另一算法。算法步骤如下：

S1 计算 $D = a_{11}a_{22} - a_{21}a_{12}$ 。

S2 如果 $D = 0$ ，则原方程组无解或者有无穷多组解；否则 ($D \neq 0$)，

$$x_1 = \frac{b_1a_{22} - b_2a_{12}}{D},$$

$$x_2 = \frac{b_2a_{11} - b_1a_{21}}{D}.$$

S3 输出计算的结果 x_1, x_2 或者无法求解信息。

从解二元一次方程组的算法可以知道，求解某个问题的算法不一定是惟一的。我们现在学习的算法不同于求解一个具体问题的方法，它有如下的要求：

(1) 写出的算法，必须能解决一类问题(例如解任意一个二元一次方程组)，并且能重复使用；

(2) 算法过程要能一步一步执行，每一步执行的操作，必须确切，不能含混不清，而且经过有限步后能得出结果。

通过求解二元一次方程组的算法，我们大家已初步理解了算法的含义和要求，与你开始时的理解是否有些不同了。下面再举个例子加以说明。

例 2 写出一个求有限整数序列中的最大值的算法。

分析：这个例子对你理解算法会有很好的帮助。你可能会觉得，求一个整数序列的最大值是很简单的事，事实上也并非简单。如果从 10 个、8 个整数中找出最大值，你一眼就能看出结果。如果从 100 个整数中找出最大值，你花点时间也能够找出，如果你要从一个 1 000 000 人的年龄序列表中找出年龄最大的一

个,要是没有算法那可就是一件困难的事了.如果在你的计算机内已有了100万人口的年龄登记表,用计算机软件转瞬间就可找出最大值.计算机能快速找出是靠软件(程序)支持,编写程序要依赖算法.如何对这个问题写出一个算法呢?

解:为了便于理解,算法步骤用自然语言叙述:

S1 先假定序列中的第一个整数为“最大值”.

S2 将序列中的下一个整数值与“最大值”比较,如果它大于此“最大值”,这时你就假定“最大值”是这个整数.

S3 如果序列中还有其他整数,重复 S2.

S4 在序列中一直到没有可比的数为止,这时假定的“最大值”就是这个序列中的最大值.

如果让你去找,你可能不会这样做,可能认为,这样太机械、太枯燥.不要忘了,我们写的是算法.算法要求“按部就班地做”,每做一步都有惟一的结果,又要求写出的算法对任意整数序列都适用,并且在有限步之后,总能得出结果.所以上面写的,符合算法要求.

下面,我们用数学语言,写出对任意3个整数 a 、 b 、 c 求出最大值的算法,再帮助你理解算法概念.

S1 $max = a$. (max 表示最大值,这个式子的意思是,假定最大值是第一个整数)

S2 如果 $b > max$, 则 $max = b$.

S3 如果 $c > max$, 则 $max = c$.

S4 max 就是 a 、 b 、 c 中的最大值.

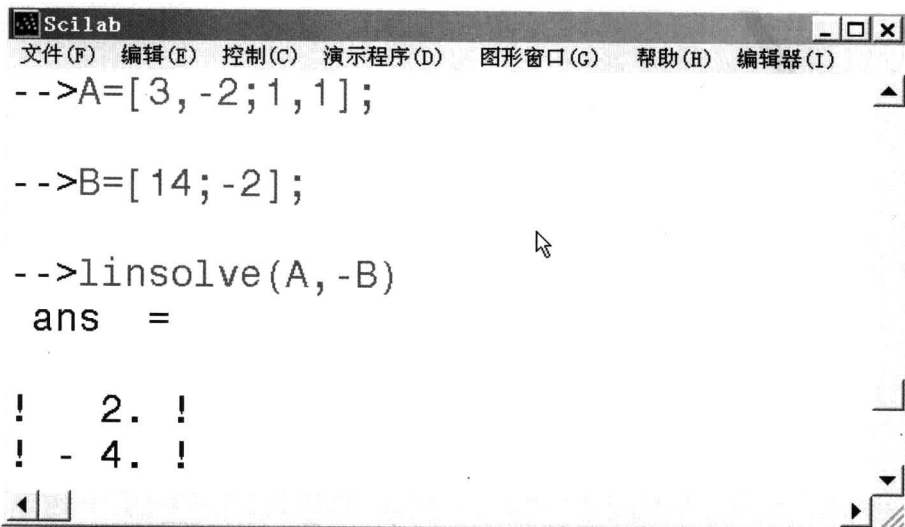
有了算法,就可编写程序.同学们现在可能还不会编程序,没关系,大部分数学计算都已由科学家编制成应用程序.大家不妨先来感受一下这些应用程序的威力,并加深对算法作用的理解.

例 3 应用 Scilab 计算指令解方程组:

$$\begin{cases} 3x - 2y = 14 \\ x + y = -2 \end{cases}$$

解:用 Scilab 程序:

打开 Scilab 程序,在界面上按下图中的格式输入两个未知数的系数和常数项:



```

Scilab
文件(F) 编辑(E) 控制(C) 演示程序(D) 图形窗口(G) 帮助(H) 编辑器(I)
-->A=[3, -2; 1, 1];

-->B=[14; -2];

-->linsolve(A, -B)
ans =

!   2. !
! - 4. !

```

得到这个方程组的解是 $x=2$, $y=-4$.

不论给出的是多少个未知数的线性方程组, 只要按上面的格式, 在 Scilab 界面上输入给定的数据, 转瞬间就会输出解答.

在计算机上能够求解方程组, 是由于计算机安装有计算软件, 而软件的核心是算法. 只要有了解决问题的算法, 不管借助的工具是纸笔、计算器, 还是计算机, 都能按照算法步骤求得相同的结果.



思考与讨论

说出你过去和现在对“算法”一词的理解.

做一做本小节的练习, 看看你是否能对算法有较正确的理解?



练习 A

1. 写出你在家烧开水过程的一个算法.
2. 写出一个能找出 a 、 b 、 c 、 d 四个数中最大值的算法.
3. 写出用高斯消去法解二元一次方程组的一个算法.
4. 用高斯消去法解下列二元一次方程组:

$$(1) \begin{cases} 3x-2y=-2 \\ 7x-y=18 \end{cases} \quad (2) \begin{cases} 2x+3y=3 \\ 6x+12y=11 \end{cases}$$



练习 B

1. 写出一个能找出 a 、 b 、 c 三个数中最小值的算法。
2. 写出一个解一元一次方程的算法。
3. 写出在下面的数字序列中，搜索数 18 的一个算法：
2 5 7 8 15 32 18 12 8 52
4. 写出两个分数乘法的算法。
5. 用 Scilab 计算指令解下列方程组：

$$(1) \begin{cases} 3x - 2y + z = -4 \\ 5x + 2y - 3z = 12 \\ x + 3x + 5y = -1 \end{cases} \quad (2) \begin{cases} x + y + z = 12 \\ 3x - 3y - z = 16 \\ x - y - z = -2 \end{cases}$$

1.1.2 程序框图

1. 程序框图的概念

通常用一些通用图形符号构成一张图来表示算法。这种图称做程序框图(简称框图)。上一小节中，用公式法解二元一次方程组的算法可用框图形象地描述如图 1-1。

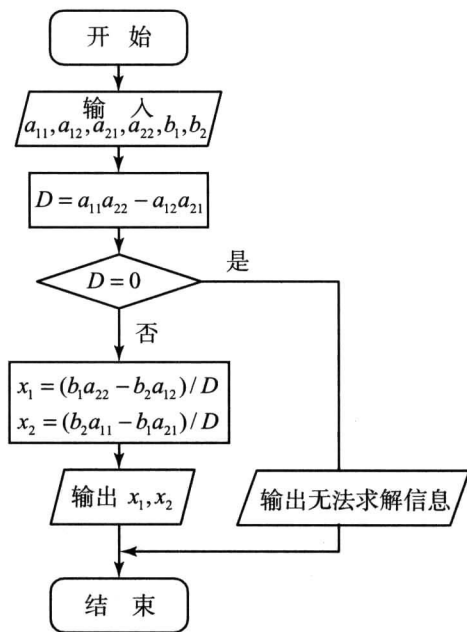
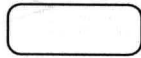
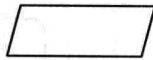
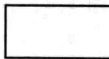


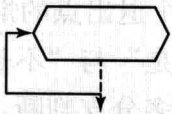

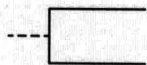


图 1-1

由此我们可以看到用框图表示算法直观、形象，容易理解。通常说“一图胜万言”，就是说用框图能够清楚地展现算法的逻辑结构。

在图 1-1 的框图中有许多图形符号和连结线，这些符号表示特定的含义，被大家普遍采用。下面列表给出一些常用的表示算法步骤的图形符号。

图形符号	名称	符号表示的意义
	起、止框	框图的开始或结束
	输入、输出框	数据的输入或者结果的输出
	处理框	赋值、执行计算语句、结果的传送
	判断框	根据给定条件判断
	流程线	流程进行的方向
	循环框	程序做重复运算
	连结点	连结另一页或另一部分的框图
	注释框	帮助理解框图

下面对图形符号再作一些说明：

起、止框是任何流程不可少的，表明程序开始和结束。输入和输出可用在算法中任何需要输入、输出的位置。例如求解方程组的框图(图 1-1)中，算法开始后第一步需要输入(给定)未知数的系数和常数项，就可把给定的数值写在输入框内，最后要给出运算的结果，


练习 B

1. 画出你每天起床过程的程序框图.

2. 画出解方程组 $\begin{cases} 3x-y=5 \\ 2x+5y=8 \end{cases}$ 的程序框图.

1.1.3

算法的三种基本逻辑结构和框图表示

我们写出的算法或画出的程序框图,一定要使大家一步步地看得清楚、明白,容易阅读.不然,写的算法乱无头绪,就很难让人阅读和理解.这就要求算法或程序框图有一个良好的结构.通过对各种各样的算法和框图进行分析和研究,证明只须用顺序结构、条件分支结构和循环结构就可表示任何一个算法.用这三种基本结构表述的算法和画出的框图,整齐美观,容易阅读和理解.下面我们分别介绍这三种基本逻辑结构.

1. 顺序结构

顺序结构描述的是最简单的算法结构,语句与语句之间,框与框之间是按从上到下的顺序进行.下面举例说明.

例 1 已知点 $P_0(x_0, y_0)$ 和直线 $l: Ax + By + C = 0$, 求点 $P_0(x_0, y_0)$ 到直线 l 的距离 d .

解: (1) 用数学语言来描述算法:

S1 输入点的坐标 x_0, y_0 , 输入直线方程的系数和常数 A, B, C ;

S2 计算 $z_1 = Ax_0 + By_0 + C$;

S3 计算 $z_2 = A^2 + B^2$;

S4 计算 $d = \frac{|z_1|}{\sqrt{z_2}}$;

S5 输出 d .

(2) 用框图来描述算法, 如图 1-3 所示.

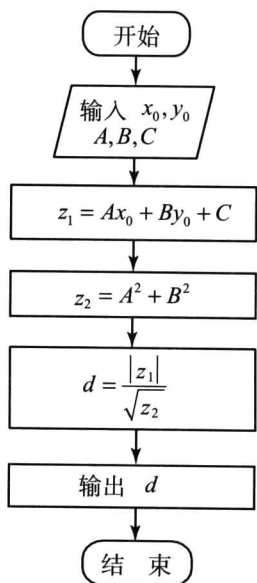


图 1-3