

丛书策划 华育文化传播公司

G

高中生

[人教B版]

GAOZHONGSHENGXUEXIZHIDA

学习指导

数学 ③

必修

辽宁师范大学出版社

高中生

[人教B版]

GAOZHONGSHENGXUEXIZHIDAO

学习指导

丛书主编 杜贵忠

本册主编 苏文捷

本册副主编 刘子军 邢长艳

本册编者 侯雪晨 刘新风 王苗苗

数学③

必修

辽宁师范大学出版社
·大连·

©杜贵忠 2007

图书在版编目(CIP)数据

高中生学习指导:人教B版.数学.3:必修/杜贵忠
主编. —大连:辽宁师范大学出版社,2007.11
ISBN 978-7-81103-706-7

I. 高... II. 杜... III. 数学课-高中-教学参考资料
IV. G634

中国版本图书馆 CIP 数据核字(2007)第 183761 号

出 版 人:程培杰
责任编辑:吕英辉 王 琦
责任校对:王媛媛
封面设计:李小曼

出 版 者:辽宁师范大学出版社
地 址:大连市黄河路 850 号
邮 编:116029
营销电话:(0411)84206854 84215261 84259913(教材)
印 刷 者:沈阳全成广告印务有限公司
发 行 者:辽宁时代华育书业发展有限公司

幅面尺寸:210mm×285mm
印 张:7.5
字 数:225 千字

出版时间:2007 年 12 月第 1 版
印刷时间:2007 年 12 月第 1 次印刷
书 号:ISBN 978-7-81103-706-7

定 价:10.50 元

编写说明

为了适应普通高中课程改革和使用新教材的需要,切实提高高中教学质量,并努力实现减轻学生的课业负担,我们组织辽宁省部分示范性高中、重点高中的知名教师,按学科编写了高中教学辅助用书《高中生学习指导》丛书。目前,完成了语文、数学、英语(两个版本)、物理、化学、生物、思想政治、历史、地理等9个学科必修教材的配套用书,共37册,供高中教师、学生选用。

丛书体例:

《高中生学习指导》按教材的章节(或单元)顺序编排,包括以下几个部分:

目标导航与知识网络:对本章节的知识结构及要点进行归纳,让学生对本章节的知识结构有个清晰的了解。

知识提要:对本节重点知识进行概括,对易混淆知识点进行讲解。

典例探源:选择典型习题或示例,对其进行规范的分析与解答,使学生掌握正确的解题思路。

学习提点:对难掌握的知识进行归纳,梳理,提示解题方法与技巧。

学海探骊:结合本课学习内容,有针对性地精选习题,体现习题的基础性、层次性、选择性。

章末检测:对本章内容进行测试,检验学生对本章知识的掌握情况。

模块综合测试:对本模块教学内容进行综合测试,考查学生对模块教学内容的掌握情况。

参考答案:对全书的学海探骊、章末检测及模块综合测试中的习题给出正确答案,对易错题进行思路点拨。

丛书特点:

与新教材紧密配合,与课程计划同步;体现课改理念,符合课程标准要求;体现教辅用书的科学性、基础性、层次性、选择性;引导学生主动探究学科知识,指导学生掌握正确的学习方法;精选习题,注意减轻学生的学习负担;充分体现名校、名师的教学经验,实现资源共享。

本册由东北育才学校编写,由苏文捷任本册主编,刘子军、邢长艳任本册副主编。

本套丛书的编写力求贴近学生学习的实际需要,有效提高学生自主学习的能力和运用所学知识分析问题、解决问题的能力。希望老师和同学们能在使用过程中,提出宝贵的补充意见和修改建议,以使本套丛书在修订后更臻完善。

杜贵忠

目 录

第一章 算法初步

1.1 算法与程序框图	2
1.1.1 算法的概念	2
1.1.2 程序框图	5
1.1.3 算法的三种基本逻辑结构和框图表示	5
1.2 基本算法语句	9
1.2.1 赋值、输入和输出语句	9
1.2.2 条件语句	13
1.2.3 循环语句	16
1.3 中国古代数学中的算法案例	19
章末检测	22

第二章 统 计

2.1 随机抽样	26
2.1.1 简单随机抽样	26
2.1.2 系统抽样	30
2.1.3 分层抽样	33
2.1.4 数据的收集	38
2.2 用样本估计总体	41
2.2.1 用样本的频率分布估计总体的分布	41
2.2.2 用样本的数字特征估计总体的数字特征	45
2.3 变量的相关性	50
2.3.1 变量间的相关关系	50
2.3.2 两个变量的线性相关	50
章末检测	56

第三章 概 率

3.1 事件与概率	61
3.1.1 随机现象	61

3.1.2 事件与基本事件空间	61
3.1.3 频率与概率	64
3.1.4 概率的加法公式	68
3.2 古典概型	71
3.2.1 古典概型	71
3.2.2 概率的一般加法公式(选学)	71
3.3 随机数的含义与应用	74
3.3.1 几何概型	74
3.3.2 随机数的含义与应用	74
3.4 概率的应用	77
章末检测	81
模块综合测试	85
参考答案	89

第一章 算法初步

目标导航

1. 本章的课标要求包括算法的概念、程序框图、基本算法语句, 阅读中国古代教学中的算法案例, 体会中国古代数学对世界数学发展的贡献.

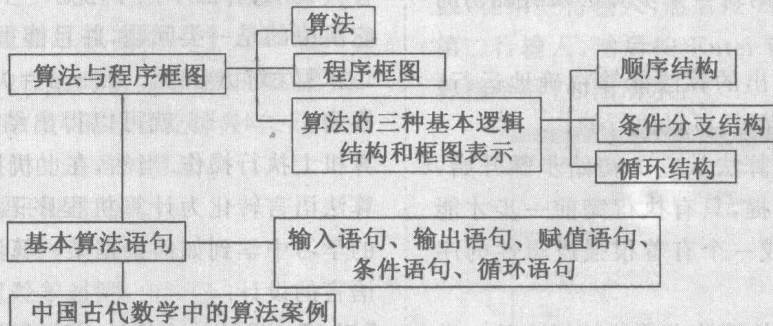
2. 算法就是解决问题的步骤, 算法也是数学及其应用的重要组成部分, 是计算机科学的基础. 在日常生活中做任何事情都有算法, 利用计算机解决问题也需要算法. 计算机可以解决多类信息处理问题, 但人们必须事先用计算机能够理解的语言, 也就是程序设计语言来详细描述解决问题的步骤, 即首先设计程序. 对复杂一些的问题, 直接写出解决该问题的程序是困难的, 首先我们要研究解决问题的算法, 再把算法转化为程序, 因此算法设计是使用计算机解决具体问题的一个极为重要的环节.

3. 通过对解决具体问题(如二元一次方程组的求解等问题)的过程与步骤的分析, 体会算法的思想, 了解算法的含义. 理解程序框图的三种基本逻辑结构: 顺序结构、条件分支结构、循环结构. 理解并掌握几种基本的算法语句——输入语句、输出语句、赋值语句、条件语句和循环语句.

4. 本章的重点是体会算法的思想, 了解算法的含义, 通过模仿、操作、探索, 经历通过设计程序框图解决问题的过程. 难点是在具体问题的解决过程中, 理解三种基本逻辑结构, 经历将具体问题的程序框图转化为程序语句的过程, 理解几种基本的算法语句.

5. 在学习算法时, 可以与计算机等信息技术课程很好地联系在一起. 同时通过计算机实现算法, 解决数学问题还可以使我们看到算法的可行性、应用性, 激发我们学习算法的积极性. 但应注意, 不能将数学中算法的学习变为计算机中的程序语言的学习或程序设计, 不能因为应用而本末倒置.

知识网络



1.1 算法与程序框图

1.1.1 算法的概念

知识提要

1. 算法的含义

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.

(1)算法不是计算方法的简称,不要认为只有“计算”才有算法.广义地说,算法就是做某一件事的步骤或程序.例如:菜谱是做菜肴的算法;洗衣机的使用说明书是操作洗衣机的算法;歌谱是一首歌曲的算法.

(2)在数学中,主要研究计算机能实现的算法,即按照某种程序步骤一定可以得到结果的解决问题的程序.比如解方程的算法、函数求值的算法等.算法一般是机械的,有时要进行大量重复计算,只要按部就班地去做,总能算出结果.通常把算法过程称为“算法机械化”.算法机械化的最大优点是它可以让计算机完成.

(3)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,也可以用算法语句给出精确的说明,或者用程序框图直观地显示算法的全貌.

2. 算法语言的特点

任何一个能够解决问题的算法都必须具备 5 个特点:

(1)有穷性:一个算法必须要保证在执行有限步骤后能够得出结果;

(2)确定性:算法的每一个步骤必须有确切的定义,不能含糊不清;

(3)可行性:设计出的算法能够精确地运行,算法中的每一个步骤都是可以执行的;

(4)逻辑性:一个算法从它的初始步骤开始,前一步是后一步的前提,只有执行完前一步才能进行下一步,从而组成一个有着很强逻辑性的序列;

(5)概括性:写出的算法必须能够解决某一类问题,并且能重复使用.

3. 算法的描述

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, ……S 是 Step(步骤)的缩写.

4. 用高斯消去法解一般的二元一次方程组

$\begin{cases} a_{11}x_1 + a_{12}x_2 = b_1 \\ a_{21}x_1 + a_{22}x_2 = b_2 \end{cases}$ 的一个算法

令 $D = a_{11}a_{22} - a_{21}a_{12}$,

若 $D = 0$, 方程组无解或有无穷多组解;

若 $D \neq 0$, 则

$$x_1 = \frac{b_1a_{22} - b_2a_{12}}{D}, x_2 = \frac{b_2a_{11} - b_1a_{21}}{D}.$$

由此可得解二元一次方程组的一个算法,其步骤如下:

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.

典例探源

【例 1】 写出解一元二次方程 $ax^2 + bx + c = 0$, ($a \neq 0$) 的一个算法.

【解】 S1 计算 $\Delta = b^2 - 4ac$.

S2 如果 $\Delta < 0$, 则方程无解; 否则 ($\Delta \geq 0$), 方程两根 $x_{1,2} = \frac{-b \pm \sqrt{\Delta}}{2a}$.

S3 输出计算结果 x_1, x_2 或无解信息.

说明: 算法不同于求解一个具体问题的方法, 它解决的是一类问题, 并且能重复使用. 在应用例 1 中算法解方程 $x^2 - 4x - 5 = 0$ 时, 将 a, b, c 具体换为 1, -4, -5 就可以得出结果了, 这易于在计算机上执行操作. 当然, 在上机操作之前还需要将算法语言转化为计算机程序语言, 我们会在后面的学习中学到如何根据某一算法进行计算机程序语言的设计.

【例 2】 设计一个算法, 求 $1 \times 3 \times 5 \times 7 \times 9$ 的值.

【解】 S1 先求 1×3 , 得到结果 3.

S2 将 S1 得到的乘积 3 再乘以 5, 得到结果 15.

S3 将 S2 得到的乘积 15 再乘以 7, 得到结果 105.

S4 将 S3 得到的乘积 105 再乘以 9, 得到结果 945, 即是最后结果.

说明:有人可能会认为, 一眼就看出答案了, 为什么还一步一步地做, 太枯燥了. 事实上, 相乘的数小、数少还能容易看得出结果, 但如果数大、数多了, 没有这样的过程和步骤就很难去解决这一问题. 最主要的是, 这样的算法有利于在计算机上执行操作.

【例 3】 设计一个算法, 使得从 10 个确定且互不相等的数中挑选出最大的一个数.

【解】 算法 1:

S1 假定这 10 个数中第一个是“最大值”.

S2 将下一个数与“最大值”比较, 如果它大于此“最大值”, 那么就用这个数取代“最大值”, 否则就取“最大值”.

S3 重复 S2.

S4 在这 10 个数中一直取到没有可以取的数为止, 此时的“最大值”就是 10 个数中的最大值.

算法 2:

S1 把 10 个数分成 5 组, 每组两个数, 同组的两个数比较大小, 取其中的较大值.

S2 将所得的 5 个较大值按 2, 2, 1 分组, 有两个数的组在组内比较大小, 取其中的较大值; 只含一个数的组不变.

S3 从剩下的 3 个数中任意取两个数比较大小, 取其中较大值, 并将此较大值与另一个数比较, 此时的较大值就是 10 个数中的最大值.

说明:算法实际上是解决问题的步骤和方法, 由于求解问题的出发点不同, 能得到不同的算法. 须注意: 不必刻意追求最优的算法, 把握算法的基本结构和程序化思想才是我们的学习重点.

【例 4】 一个人带三只狼和三只羚羊过河, 只有一条船, 同船可以容纳一个人和两只动物. 没有人在的时候, 如果狼的数量不少于羚羊的数量, 狼就会吃掉羚羊. 请你设计一个最优的过河的算法.

【解】 S1 人带两只狼过河.

S2 人自己返回.

S3 人带一只羚羊过河.

S4 人带两只狼返回.

S5 人带两只羚羊过河.

S6 人自己返回.

S7 人带两只狼过河.

S8 人自己返回.

S9 人带一只狼过河.

说明:在日常生活中无论我们做什么事情, 都是按照一定的规则, 一步步地进行的, 都离不开算法. 在算法的表达中可以用自然语言描述. 在设计算法时, 应当先找到解决问题的方案, 再把它细化为一步连接一步的步骤, 从而设计出算法. 如果要设计出计算机的程序语言, 则要用相关的数值来代表其中的具体操作.

【例 5】 写出判断圆 $(x-x_0)^2+(y-y_0)^2=r^2$ 和直线 $Ax+By+C=0$ 的位置关系的算法.

【解】 S1 输入圆心的坐标 (x_0, y_0) , 直线方程的系数 A, B, C 和半径 r .

$$S2 \text{ 计算 } d = \frac{|Ax_0 + By_0 + C|}{\sqrt{A^2 + B^2}}.$$

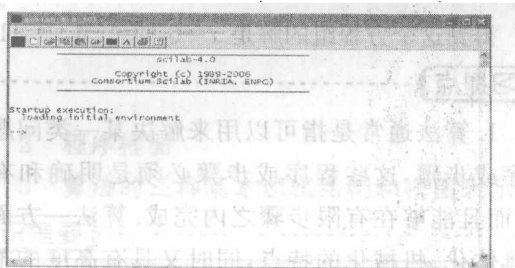
S3 如果 $d > r$, 则直线与圆相离; 如果 $d = r$, 则直线与圆相切; 如果 $d < r$, 则直线与圆相交.

说明:我们所学的数学公式, 例如四则运算就是已经设计好的算法. 我们在设计新问题的算法时, 如果有公式可用, 应尽可能地利用公式.

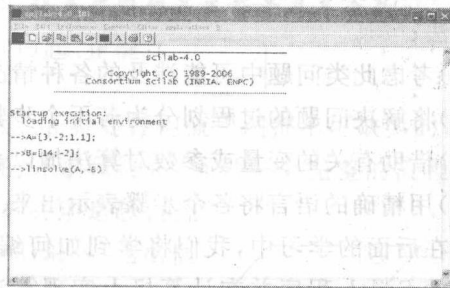
【例 6】 应用 Scilab 计算指令解方程组:

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

【解】 打开 Scilab 程序, 初始界面如图所示:



在界面上按下图的格式输入两个未知数的系数(在第一行输入, 然后按 Enter 键)和常数项(在第二行输入, 然后按 Enter 键), 在第三行输入运算指令后, 如图所示:



按 Enter 键,得到这个方程组的解是 $x=2$,
 $y=-4$.

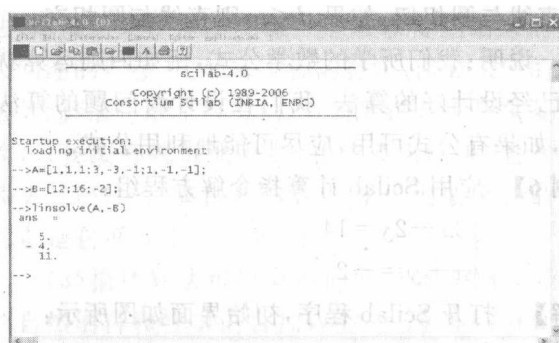


说明:不论给出的是多少个未知数的线性方程组,只要按上面的格式,在 Scilab 界面上输入给定的数据,转瞬间就会输出解答.

如:应用 Scilab 计算指令解方程组

$$\begin{cases} x+y+z=12 \\ 3x-3y-z=16 \\ x-y-z=-2 \end{cases}$$

按下图的格式输入数据:



得到这个方程组的解是 $x=5, y=-4, z=11$.

学习要点

1. 算法通常是指可以用来解决某一类问题的程序或步骤,这些程序或步骤必须是明确和有效的,而且能够在有限步骤之内完成.算法一方面具有具体化、机械化的特点,同时又具有高度的抽象性、概括性和精确性,是数学及其应用的重要组成部分,又是计算科学的重要基础.

2. 对于一个问题,设计算法时应注意:

(1)认真分析问题,思考解决此类问题的一般数学方法;

(2)考虑此类问题中可能涉及的各种情况;

(3)将解决问题的过程划分为若干个步骤;

(4)借助有关的变量或参数对算法加以表述;

(5)用精确的语言将各个步骤表示出来.

3. 在后面的学习中,我们将学到如何编写一个算法的 Scilab 程序并在计算机上实现算法.希

望有条件的同学能学会使用 Scilab 软件,在应用中领会它的功能. Scilab 科学计算软件可以从网上免费下载,现版本是 4.0.

学海探骊

1. 下列关于算法的说法中,正确的有 ()

- ①求解某一类问题的算法是唯一的 ②算法必须在有限步操作之后停止 ③算法的每一步操作必须是明确的,不能有歧义或模糊 ④算法执行后一定产生确定的结果

A. 1 个 B. 2 个 C. 3 个 D. 4 个

2. 早上从起床到出门上学需要洗脸刷牙(5 min)、刷水壶(2 min)、烧水(8 min)、泡面(3 min)、吃饭(10 min)、听广播(8 min)几个步骤,下列选项中你认为最优的一个算法为

()

A. S1 洗脸刷牙; S2 刷水壶; S3 烧水; S4 泡面; S5 吃饭; S6 听广播.

B. S1 刷水壶; S2 烧水的同时洗脸刷牙; S3 泡面; S4 吃饭; S5 听广播.

C. S1 刷水壶; S2 烧水的同时洗脸刷牙; S3 泡面; S4 吃饭的同时听广播.

D. S1 吃饭的同时听广播; S2 泡面; S3 烧水的同时洗脸刷牙; S4 刷水壶.

3. 下面给出的是一个已打乱顺序的“找出 a, b, c, d 四个数中的最小值”的算法步骤,正确的序号为

()

- ① $\min = a$ (符号“ $\min$ ”表示最小值) ② 输出 $\min$ ③ 如果 $d < \min$, 则 $\min = d$ ④ 如果 $b < \min$, 则 $\min = b$ ⑤ 输入 a, b, c, d 四个数

⑥ 如果 $c < \min$, 则 $\min = c$

A. ⑤①④⑥③② B. ⑤②④③⑥①

C. ⑤⑥③④①② D. ⑤①④⑥②③

4. 下面 Scilab 语言程序的功能是 ()

```
A=[1,3;2,-1];
B=[1;-5];
linsolve(A,-B)
```

A. 解方程组 $\begin{cases} x+3y=1 \\ 2x-y=-5 \end{cases}$

B. 解方程组 $\begin{cases} x+2y=1 \\ 3x-y=-5 \end{cases}$

C. 计算 $(1+3) \times 1 + (2-1) \times (-5)$

D. 计算 $(1+3) \times 1 - (2-1) \times (-5)$

5. 已知一个学生的语文成绩为 89, 数学成绩为

96, 外语成绩为 99, 求他的总分和平均成绩的一个算法为:

S1 取 $A=89, B=96, C=99$.

S2 $\underline{\hspace{2cm}}$ (1).

S3 $\underline{\hspace{2cm}}$ (2).

S4 输出 D, E .

6. 应用高斯消去法解二元一次方程组

$$\begin{cases} x-2y=-1 \\ 2x+y=1 \end{cases}$$

7. 写出在数字序列 15, 7, 18, 35, 27, 48, 98, 83, 68 中搜索出数字 27 的一个算法.

8. 已知直角三角形两直角边 a, b , 写出求斜边上的高的一个算法.

9. 写出一个求解任意二次函数 $y=ax^2+bx+c$ ($a \neq 0$) 的最值的算法.

10. 写出解不等式 $ax+b>0$ 的算法.

11. 在 9 枚金币中有一枚略轻的是假金币, 请设计一个算法, 仅用天平 (不用砝码) 把假金币找出来.

12. 2006 年 12 月 26 日, 因为海底地震, 上海至洛杉矶的海底光缆在某一处发生断路, 请你设计一个检测故障点的方案.

13. 如果直线 l 与直线 $x+y-1=0$ 关于 y 轴对称, 求直线 l 的方程, 请设计一个解决该问题的算法.

1.1.2 程序框图

1.1.3 算法的三种基本逻辑结构和框图表示

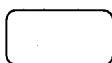
知识提要

1. 程序框图

程序框图也叫流程图, 是人们将思考的过程和工作的顺序进行分析、整理, 用规定的文字、符号、图形的组合加以直观描述的方法.

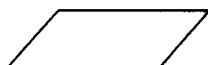
2. 程序框图的基本符号及其含义

(1) 起、止框:



起、止框是任何流程图都不可缺少的, 它表明程序的开始和结束, 所以一个完整的流程图的首末两端必须是起、止框.

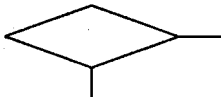
(2) 输入、输出框:



表示数据的输入或者结果的输出,它可用在算法中的任何需要输入、输出的位置.

(3)处理框: 

它是用来赋值、执行计算语句、传送运算结果的图形符号.

(4)判断框: 

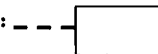
判断框一般有一个入口和两个出口,有时也有多个出口,它是唯一的具有两个或两个以上出口的图形符号.在只有两个出口的情形中,通常都分成“是”与“否”(也可用“Y”与“N”)两个分支,构成一个选择性结构.在必修3(B版)教材中我们只使用有两个出口的判断框.

(5)流程线: 

表明流程进行的方向.

(6)连结点: 

用于连结另一页或另一部分的框图.

(7)注释框: 

用于帮助理解框图.

3. 画程序框图的规则

(1)使用标准的框图的符号.

(2)框图一般按从上到下、从左到右的方向画.

(3)除判断框外,大多数流程图符号只有一个进入点和一个退出点.判断框是具有超过一个退出点的唯一符号.

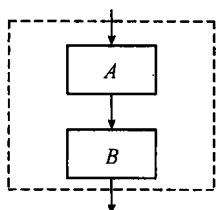
(4)判断框分两大类,一类判断框是“是”与“否”两分支的判断,而且有且仅有两个结果;另一类是多分支判断,有几种不同的结果.

(5)在图形符号内描述的语言要非常简练清楚.

4. 算法的三种基本逻辑结构和框图表示

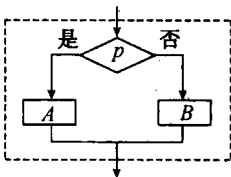
(1)顺序结构

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.如图所示虚线框内的程序框图为顺序结构的示意图.其中A和B两个框是依次执行的,在执行完A框所指定的操作后,必然接着执行B框所指定的操作.

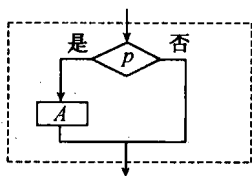


(2)条件分支结构

在一个算法中,经常会遇到一些条件的判断,算法的流程根据条件是否成立有不同的流向.由这种根据作出的判断,再决定执行哪一种操作的结构称为条件分支结构.如图所示的虚线框内的程序框图为条件分支结构的示意图.此结构中包含一个判断框,根据给定的条件 p 是否成立而选择执行A框或B框.请注意,无论条件 p 是否成立,只能执行A框或B框之一,不能既执行A框又执行B框,也不能A框、B框都不执行.

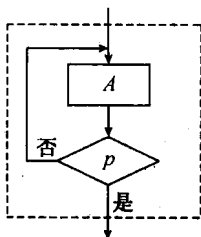


无论走哪一条路径,在执行完A框或B框之后就脱离本条件分支结构.A或B两个框中,可以有一个是空的,即不执行任何操作,如图所示.



(3)循环结构

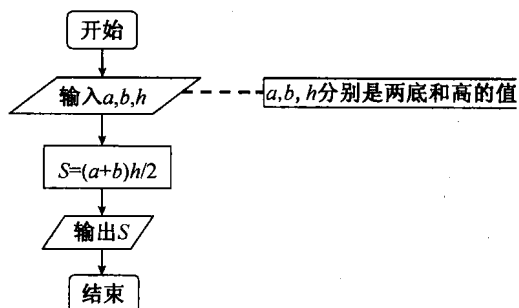
根据指定条件决定是否重复执行一条或多条指令的控制结构称为循环结构.即从某处开始,按照一定的条件反复执行某一处理步骤.如图所示的虚线框内的程序框图就是一种循环结构的示意图.它的功能是先执行A框,然后判断给定的 p 条件是否成立,如果 p 条件不成立,则再执行A框,然后再对 p 条件作判断,如果 p 条件仍然不成立,又执行A框……如此反复执行A框,直到给定的 p 条件成立为止,此时不再执行A框,脱离本循环结构.



典例探源

【例1】 已知梯形的两底为 a, b ,高为 h ,画出计算梯形面积的程序框图.

【解】 程序框图如图所示:

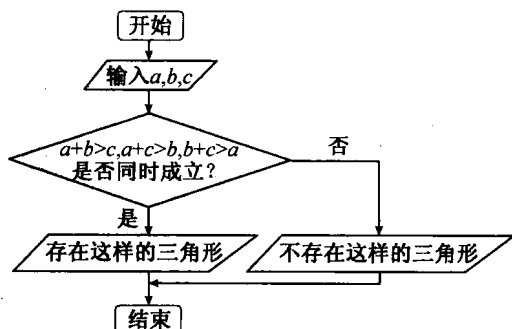


说明:这是一个简单的问题,将数值代入公

式,最后输出结果,只用顺序结构就可以表示该算法.此图的输入框旁边加了一个注释框——,它的作用是对框中的数据或内容进行说明,注释框可以出现在任何位置.

【例2】 任意给定3个正实数,判断分别以这3个数为三边边长的三角形是否存在,画出这个算法的程序框图.

【解】 程序框图:



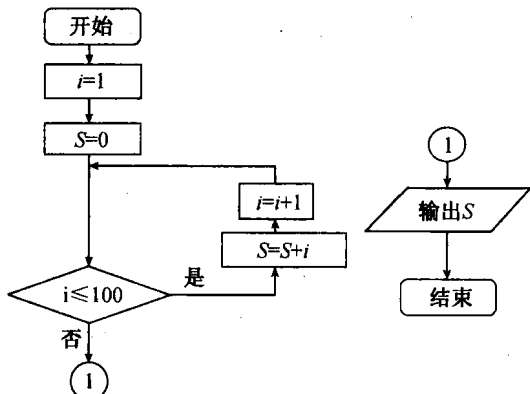
说明: 判断分别以这3个数为三边边长的三角形是否存在,只要验证这3个数当中任意两个数的和是否大于第3个数,这就需要用到条件分支结构.

【例3】 设计一个计算 $1+2+3+\dots+100$ 的算法,并画出程序框图.

【解】 算法:

- S1 $i=1$.
- S2 $S=0$.
- S3 如果 $i \leq 100$, 则执行 S4, S5; 否则, 执行 S6.
- S4 $S=S+i$.
- S5 $i=i+1$.
- S6 输出 S .

程序框图:



说明: (1) 在程序框图中 $S=S+i \rightarrow i=i+1$ 部分被重复执行,我们称这一部分为循环体.每循环一次, S 和 i 都要发生变化,这两步要重复计算100次.

(2) 这里的 S 代表的是“和”,是累加变量,它

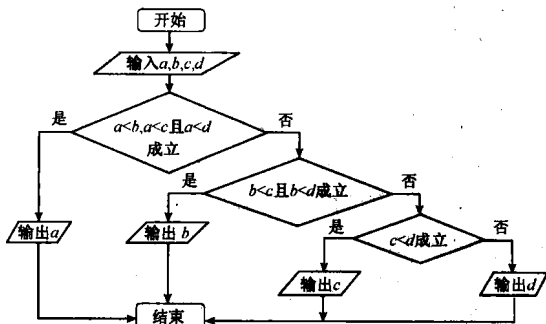
的值随着 i 的变化而变化,“ $S=S+i$ ”并不是 S 与 $S+i$ 相等,而是指将 S 的原值加 i 再赋给 S , S 的值就变为“原值加 i ”,若 S 原值是“0”,则加1后就变为“1”.

(3) i 代表“计数变量”,它的值不停地变化(由1到100),从而实现累加的目的,同样“ $i=i+1$ ”并不代表 i 与 $i+1$ 相等,而是指将 $i+1$ 赋给 i .

(4) 循环结构要有终止循环体的条件,不能无休止地运算下去,循环结构中一定包含条件结构.如本循环结构先判断 $i \leq 100$ 是否成立,若成立,执行循环体;若不成立,则终止循环.

(5) 如果一个程序框图要分开来画,则需要断开处画上连结点,并标出连结号码.在该程序框图中有两处以 $\bigcirc$ 为标志的连结点(在连结点图中写上“1”),它表示这两个点是互相连结在一起的.实际上它们是同一个点,只是在一个页面中画不下才分开画.用连结点,可以避免流程线的交叉或过长,使流程图清晰.

【例4】 根据所给程序框图回答下列问题:



(1) 若输入四个数 5, 9, 4, 6, 则最终输出的结果是什么?

(2) 该程序框图是为什么问题而设置的? 写出相应的算法.

【解】 (1) 若输入四个数 5, 9, 4, 6, 则最终输出的结果是 4.

(2) 该程序框图所对应的问题是求 a, b, c, d 四个数中的最小数. 算法如下:

- S1 输入 a, b, c, d .
- S2 如果 $a < b, a < c, a < d$, 则输出 a ; 否则, 执行 S3.
- S3 如果 $b < c, b < d$, 则输出 b ; 否则, 执行 S4.
- S4 如果 $c < d$, 则输出 c ; 否则, 执行 S5.
- S5 输出 d .

说明: 将这类较为复杂的程序框图所表达的算法翻译成自然语言时应先取一个特例(如本例中的特殊数值 4, 9, 5, 6 参与运算), 通过运算能使

我们进一步明确该程序框图所表达的算法实质,从而使我们更加清晰地将它翻译成自然语言.

学习提示

1. 条件分支结构主要用在一些需要依据条件进行判断的算法中,如分段函数的求值,数据的大小关系等问题中,常常用到条件分支结构来设计算法.

2. 循环结构主要用在一些有规律的重复计算的算法中,如累加求和、累乘求积等问题常常需要用循环结构来设计算法.

3. 在循环结构中,要注意根据条件,设计合理的计数变量、累加变量等,特别要注意循环结构中条件的表述要恰当、精确,以免出现多一次循环或少一次循环的情况.

学海探秘

1. 下列关于程序框图的说法正确的有 ()

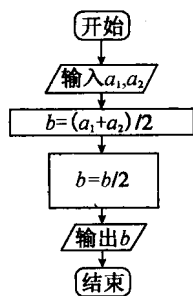
- ① 程序框图只有一个入口,也只有一个出口
- ② 程序框图中的每一部分都应有一条从入口到出口的路径通过它
- ③ 程序框图中的循环可以是无尽循环
- ④ 条件分支结构的程序框图中的两条路径可以同时执行

- A. ①②③ B. ②③
C. ①②④ D. ①②

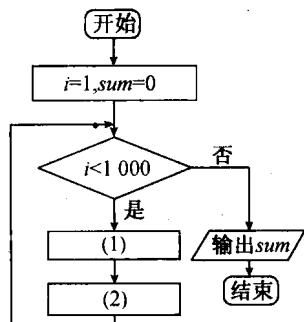
2. 要解决下面的问题,只用顺序结构画不出其程序框图的是 ()

- A. 利用公式 $1+2+3+\cdots+n=\frac{n(n+1)}{2}$, 计算 $1+2+3+\cdots+10$ 的值
- B. 已知圆的面积 S , 求其周长
- C. 给定一个数, 求其绝对值
- D. 求函数 $f(x)=x^2-2x+2$ 的函数值

3. 图中所示的是一个算法的程序框图, 已知 $a_1=3$, 输出的 $b=7$, 则 a_2 的值是 _____.



(第3题)



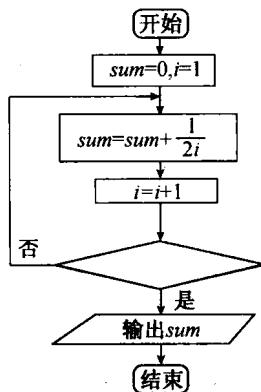
(第4题)

4. 求 1~1 000 内所有奇数的和, 根据条件把流程图补充完整.

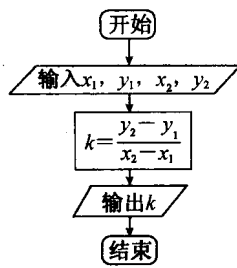
(1) 处填 _____

(2) 处填 _____

5. 如图给出的是计算 $\frac{1}{2} + \frac{1}{4} + \frac{1}{6} + \cdots + \frac{1}{20}$ 的一个程序框图, 其中判断框内应填入的条件是 _____.



(第5题)



(第6题)

6. 已知两点 $(x_1, y_1), (x_2, y_2)$, 求过两点的直线的斜率. 用程序框图表示解决这个问题一个算法如图所示, 这个程序框图对吗? 如果有错误, 请你画出一个正确的程序框图.

7. 设 x 为一个正整数, 规定如下运算: 若 x 为奇数, 则 $y=3x+2$; 若 x 为偶数, 则 $y=5x$, 求 y 的值. 写出算法, 并画出程序框图.

8. 已知三个实数 a, b, c , 试给出一个确定三个数最大值的程序框图.

9. 已知函数 $y = \begin{cases} -1, & x > 0 \\ 0, & x = 0 \\ 1, & x < 0 \end{cases}$, 写出求该函数值的

算法并画出相应的程序框图.

10. 已知 $a_1 = 1, a_{n+1} = a_n + 1 (n \in \mathbf{N}_+)$, 画出输入 n 求 a_n 的程序框图.

11. 某住宅小区的物业部门每月向居民收取卫生费, 计费方法是: 3 人及 3 人以下的住户, 每户收取 5 元; 超过 3 人的住户, 每超出 1 人加收 1.2 元. 设计一个算法, 根据输入的人数, 计算应收取的卫生费, 并画出程序框图.

12. 画出求 $2^1 + 2^2 + 2^3 + \cdots + 2^{100}$ 的值的程序框图.

13. 画出使 $1 + 2 + 3 + \cdots + n > 2\,007$ 成立的最小自然数 n 的值的程序框图.

14. 写出求 $2 + \frac{1}{2 + \frac{1}{2 + \frac{1}{2 + \frac{1}{2 + \frac{1}{2}}}}}$ 的值的的一个算

法, 并画出程序框图.

1.2 基本算法语句

计算机完成任何一项任务都需要算法, 但是, 我们用自然语言或程序框图描述的算法, 计算机是无法“看懂”的. 因此还需要将算法翻译成计算机能够理解的计算机程序语言.

程序语言有很多种, 如 Basic、Foxbase、C 语言、C++、J++、VB 等. 为了实现算法中的三种基本的逻辑结构: 顺序结构、条件分支结构和循环结构, 各种程序语言中都包含下列基本的算法语句: 输入语句、输出语句、赋值语句. 条件语句和循环语句.

下面我们来学习 Scilab 程序语言中的一些基本的算法语句, 用这些语句可以编写出 Scilab 应用程序, 实现算法思想. 建议大家多进行上机练习, 这样可以更好地理解这些语句的结构, 体会用

自行设计的 Scilab 程序解决一些数学问题带来的成就感.

1.2.1 赋值、输入和输出语句

知识提要

1. 赋值语句

(1) 在表述一个算法时, 经常要引入变量, 并赋给该变量一个值. 用来表明赋给某一个变量一个具体的确定值的语句叫做赋值语句. 在算法语句中, 赋值语句是最基本的语句.

(2) 赋值语句的一般格式是: 变量名 = 表达式. 赋值语句中的“=”号叫做赋值号.

(3) 赋值语句的作用是: 先计算出赋值号右边表达式的值, 然后把该值赋给赋值号左边的变量, 使该变量的值等于表达式的值.



2. 输入语句

(1) 在某些算法中,变量的初值要根据情况经常改变.一般我们把程序和初始数据分开,每次算题时,即使初始数据改变,也不必改变程序部分,只要每次程序运行时,输入相应的数据即可.这个过程在程序语言中,用“输入语句”来控制.

(2) 在 Scilab 中的输入语句之一是“input”.

输入语句的一般格式是:

变量名=input(“提示信息”).

当需输入单个或多个字符时,其格式是:

变量名=input(“提示信息”,“string”).

(3) 使用输入语句的意义在于:在编写程序中可以把程序和初始数据分开,达到用程序解决一类问题的目的.也就是说在程序中用字母(变量)代替数,在解决具体问题时,对变量赋值.

3. 输出语句

(1) 任何求解问题的算法,都要把求解的结果“输出”.由此可知,任何程序语言也必须有“输出语句”来控制输出.不同的程序语言都有自己的输出语句和表现形式,但功能是一样的,就是以某种形式把求解结果“输出”出来.

(2) 在 Scilab 中有各种输出语句,如 print, write, format, printf, disp. 在教材中,我们仅对“print”和“disp”语句加以说明.

输出语句“print”的一般格式是:

print(%io(2),表达式)

典例探源

【例 1】 判断下列给出的赋值语句、输入语句和输出语句是否正确.请说明理由.

(1) 赋值语句 $2=A$;

(2) 赋值语句 $M=N=6$;

(3) 输入语句 $a=\text{input}(\text{"chinese"})$;

(4) 输出语句 $\text{print}(\%io(2),a;b;c)$.

【解】 (1) 错,赋值语句中“=”左边只能是变量名,而不能是表达式或数值.

(2) 错,一个赋值语句只能给一个变量赋值,不能出现两个或多个“=”.

(3) 正确,输入语句 $a=\text{input}(\text{"chinese"})$ 中,真正起作用的是 $a=\text{input}()$,它将键盘输入的数值赋给 a ,括号中的“chinese”仅仅是起提示作用.

(4) 错,print 语句变量使用“,”,而不是“;”. $\text{print}(\%io(2),a,b,c)$ 才是正确的表示.

说明:在写赋值语句、输入语句和输出语句时,要严格按照这三种语言的要求来写.有很多微

小的细节要留心,计算机是严格按照语句来执行的,一个字母、一个标点符号的错误都会使整个程序出错.

【例 2】 分别写出下面运算输出的结果.

(1) $a=5$;

$b=3$;

$c=(a+b)/2$;

$d=c*c$;

$\text{print}(\%io(2),d)$

(2) $a=1$;

$b=2$;

$c=a+b$;

$b=a+c-b$;

$\text{print}(\%io(2),a,b,c)$

(3) $a=10$;

$b=20$;

$c=30$;

$a=b$;

$b=c$;

$c=a$;

$\text{print}(\%io(2),a,b,c)$

【解】 (1) 16

语句 $c=(a+b)/2$ 是将 a, b 和的一半赋给 c ,语句 $d=c*c$ 是将 c 的平方赋给 d ,最后输出 d 的值.

(2) 3, 2, 1

语句 $c=a+b$ 是将 a, b 的和赋给 c ,语句 $b=a+c-b$ 是将 $a+c-b$ 赋给了 b ,最后输出 c, b, a 的值.

(3) 20, 30, 20

经过语句 $a=b$ 后, a, b, c 的值是 20, 20, 30. 经过语句 $b=c$ 后, a, b, c 的值是 20, 30, 30. 经过语句 $c=a$ 后, a, b, c 的值是 20, 30, 20.

说明: (1) $\text{print}(\%io(2), \text{表达式})$ 中的表达式指程序要输出的数据,输出语句可以输出常量、变量或表达式的值,例如 $\text{print}(\%io(2), B)$, $\text{print}(\%io(2), 4*3)$ 等.

(2) $\text{print}(\%io(2), a, b, c)$ 在屏幕上输出的顺序是 c, b, a . 如果要输出 a, b, c 则应写成 $\text{print}(\%io(2), c, b, a)$.

(3) $\text{print}(\%io(2), a, b, c)$ 中的 io 表示 input-output(输入-输出), $\%io(2)$ 表示在屏幕上输出.

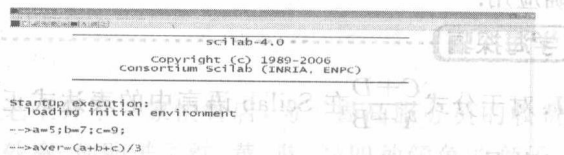
【例 3】 任给三个变量 a, b, c 赋值,计算这三个数的算术平均数.

【解】 $a=\text{input}(\text{"a="})$;

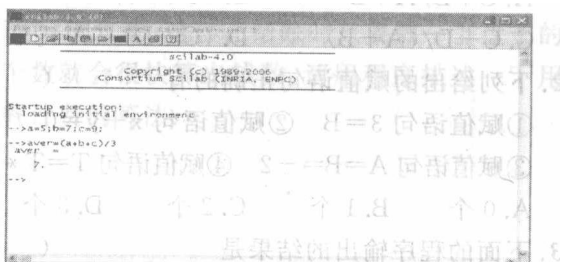
$b=\text{input}(\text{"b="})$;

```
c=input("c=");
aver=(a+b+c)/3;
print(%io(2),aver)
```

说明:Scilab 自带了一些运算程序,如果只是想计算某三个确定的数,如 5,7,9 的平均数,可以直接在 Scilab 窗口界面内输入:



按“Enter”键,界面出现运算结果:



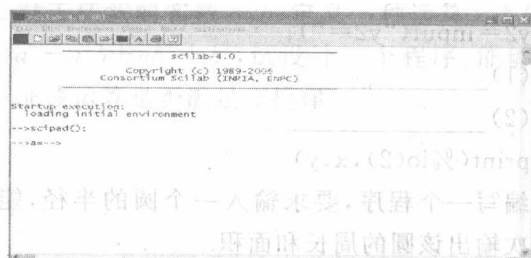
但在某些算法中,变量的初值要根据情况经常改变.一般我们把程序和初始数据分开.每次算题时,即使初始数据改变,也不必改变程序部分,只要每次程序运行时,输入相应的数据即可.本例用“input”语句来控制输入,“print”语句来控制输出.

在 Scilab 界面下的菜单栏中单击“editor”,就打开了 Scilab 的文本编辑器.

我们可以在文本编辑器中写出该程序(需要注意的是,要在英文半角状态下输入标点符号,否则计算机将不识别).

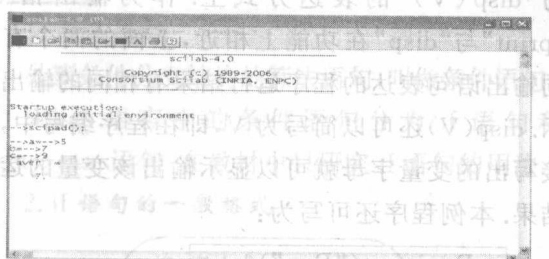
把程序保存在一个文件中,例如“d:\suanfa\aver. sci”.

在 Scilab 的文本编辑器内单击菜单栏中“Execute”下的“Load into Scilab”,立即会在 Scilab 界面内运行:



在界面内先执行了输入语句“a=input(“a=”)”,提示你要在 a= 的提示符后面输入 a 值.这时,你可以从键盘输入赋给 a 的一个数,如 5,再按“Enter”键,程序继续运行,以同样的方式

输入赋给 b 和 c 的数,如 7,9. 按“Enter”键,最后的界面出现:



【例 4】 已知圆锥底面半径和高,设计一个程序,求圆锥的体积.(锥体的体积公式: $V=\frac{1}{3}\times$ 底面积 $\times$ 高.在程序中用 %pi 表示圆周率 π)

【解】 程序为:

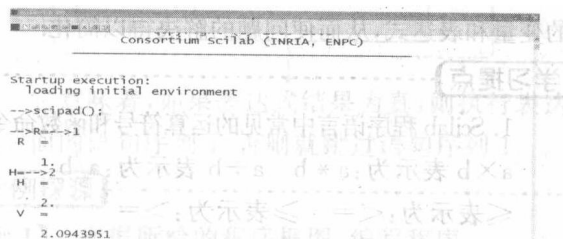
```
R=input("R=");
H=input("H=");
M= R^2;
V=%pi * M * H/3;
print(%io(2),V)
```

该程序的另一种书写是:

```
R=input("R=")
H=input("H=")
M= R^2;
V=%pi * M * H/3;
disp(V)
```

说明:注意两个程序中细微的差别:

第一处差别在“R=input(“R=”)”等语句的后面有无分号. “R=input(“R=”)”(该语句后无分号)会在程序运行时在屏幕上显示执行该语句的运行结果:



“R=input(“R=”)”(该语句后有分号)则不在程序运行时在屏幕上显示执行该语句的运行结果:

