清流一中信息技术

校本课程（Pascal）教案

第1章  计算机的发展与应用

1.1  计算机发展简史

 1.1.1  第一台电子计算机的诞生

1946年，世界上第一台数字式电子计算机由美国宾夕法尼亚大学的物理学家约翰·莫克利（John Mauchly）和工程师普雷斯伯·埃克特（PresperEckert）领导研制成功，取名为ENIAC（Electronic Numerical Integrator And Calculator）。它可以在1秒进行5000次加减运算，3毫秒便可以进行一次乘法运算，能存储20个字长为10位的十进制数。

1.1.2  计算机发展的几个阶段

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|   | 第一代（1946~1958） | 第二代（1958~1964） | 第三代（1964~1975） | 第四代（1975~至今） |
| 主要电子器件 | 电子管 | 晶体管 | 中小规模集成电路 | 大规模/超大规模集成电路 |
| 内存 | 汞延迟线 | 磁芯存储器 | 半导体存储器 | 半导体存储器 |
| 外存储器 | 穿孔卡片、纸带 | 磁带 | 磁带、磁盘 | 磁盘、光盘等大容量存储器 |
| 处理速度（指令数/秒） | 几千条 | 几百万条 | 几千万条 | 数亿条以上 |

第四代计算机以大规模集成电路VLSI为计算机的主要功能部件，用16K、64K或集成度更高的半导体存储器作为主存储器。

新机器要继承原有机器指令系统中的全部指令，使得同一系列计算机的指令系统越来越复杂，这些计算机被称为“复杂指令系统计算机”，简称CISC。

日趋庞大的指令系统降低了系统性能，而机器中最常常的是一些简单指令，这些指令仅占指令系统中指令总数的20%，1975年提出了精简指令系统，简称RISC。

冯·诺依曼型计算机是采用存储程序方式进行工作。

研制中的第五代计算机，是一种智能计算机，具有自动识别自然语言、图形、图像有能力，具有理解和推理能力，具有知识获取、知识更新能力。研制朝两个方向努力：①创建非冯·诺依曼式语言，LISP、PROLOG、F.P.；②创建以人脑神经系统处理信息的原理为基础的非冯·诺依曼式的计算机模型，生物计算机、光子计算机、量子计算机。

1.2  计算机对现代社会的影响

1.2.1  计算机应用概述

1、科学计算：利用计算机解决科学研究和工程技术中所提出的复杂的数学问题。

2、数据处理：利用计算机对所获取的信息进行记录、整理、加工、存储和传输等。

3、人工智能：利用计算机来模仿人类的智力活动。

4、自动控制：利用计算机对动态的过程进行控制、指挥和协调。

5、计算机辅助设计和制造：利用计算机来辅助工程师们设计和制造产品。

1.2.2  由信息高速公路热引发的全球信息化浪潮

在现代，能源、材料与信息是社会发展的三大支柱。

以蒸汽机为中心的动力革命，使人类的体力劳动大大得到了解放；以电子信息技术为先导、以计算机和现代通信产业为代表的信息革命使人类的脑力劳动得到解放。

社会发展到每一阶段，都需要有相应的基础设施作支持。在农业社会，田地是社会的基础设施；19世纪进行的工业革命时代，很多国家通过建设高速公路和铁路等交通运输方面的基础设施取得了经济飞速发展；而在信息化时代，信息高速公路就自然成为国家重要的基础设施。

1993年美国国家信息基础设施计划提出了一个具有大容量、高速度的电子数据传递系统。该系统由通信网络、信息设备、信息资源和信息环境组成。

1.2.3  我国的“三金”工程

1、金桥工程

即国家公用经济信息通信网络工程，是我国信息化建设的基础设施之一。

2、金关工程

是金桥工程的起步工程之一。

3、金卡工程

即电子货币工程，是金融电子化和商业流通现代化的重要组成部分。

语言是交流的工具，人与人交流要用语言，其它动物之间交流也有特定的动物的语言，人与计算机交流就要用到一些符号，这些符号的集合形成了计算机的语言。

计算机语言有一个发展过程,从机器语言开始,然后是汇编语言,现在则使用高级语言.机器语言和汇编语言属于低级语言,计算机容易懂,但人理解起来困难.高级语言人比较容易理解,因此使得计算机得以普及,当然这是指60年代.实际上,随着应用软件的发展,现在人们连语言也不用学了,也可以使用计算机,而且使用的很好.

当然,高级语言,人虽然容易懂,但计算机又不懂,因此需要翻译.所以高级语言都提供了编译系统或解释系统或集成的开发系统,对高级语言进行翻译,使得计算机能够识别并运行程序.高级语言还在发展,比如面向对象的语言,可视化的语言,网络开发语言等.

机器语言和汇编语言也可以称为低级语言.不要认为有了高级语言,低级语言就可以不用了.恰恰相反,在有些领域,低级语言更有用武之处,如控制领域,电子信息领域.因为这些领域对语言运行的速度和效率更感兴趣,这正是低级语言的特点.

上述语言都称为形式化的语言,与之相对应的是自然语言,也就是人使用的语言.人与计算机之间能否通过自然语言交流呢?这个恐怕很难.因为按照现在计算机的原理,计算机只有执行程序才能够工作,语言就是用于编出程序来,指挥计算机的工作,没有程序,计算机就不能工作.除非在人与计算机交流时,计算机能把人的语言翻译成程序,否则,人与计算机之间永远也不能实现用自然语言进行交流.而目前这是不太可能的.要么就是改变现在计算机的结构,改变现在计算机的运行模式,那样的话,计算机就要真正的实现更新换代了.

第1课  认识PASCAL语言

**请参阅《全国信息学奥林匹克联赛培训教程（一）》P87**

**第5章** **程序设计语言**

**5.1** **程序设计语言基础**

**5.1.1** **程序设计语言的组成**

PASCAL语言是一种高级语言。

让我们先来看一个PASCAL程序，通过这个程序了解PASCAL的规则。

**例L1\_1**已知半径，求圆周长和面积的程序

PROGRAM circle(input,output); (\* 第1行：程序首部  \*)

{已知半径求圆周长和面积}(\*  第2行：注释 \*)

  CONST(\*第3行：常量说明\*)

PI=3.14159;

  VAR(\*第5行：变量说明\*)

r,l,s:real;

  BEGIN(\*第7行：语句部分\*)

read(r);  (\*第8行：输入语句\*)

l:=2\*PI\*r;  (\*第9行：赋值语句\*)

s:=PI\*r\*r;

write(r,l,s);  (\*第11行：输出语句\*)

  END. (\*第12行：语句部分以END.结束\*)

程序的第一行为程序首部，每个PASCAL程序都必须以PROGRAM 开头；circle是本程序的名字，每个程序的名字可以不同；圆括号里的内容称为程序的参数，程序参数指明程序与外部联系的文件名，input是标准输入文件，例如键盘打字机，output是标准输出文件，指屏幕显示器或打印机，为了读入数据和输出结果，必须写上文件参数input和output。

从第三行到最后一行为程序的分程序。分程序一般由说明部分和语句部分构成。

说明部分可以包括多种类型的说明。此例包括以CONST开始的常量说明和以VAR开始的变量说明，常量说明必须以CONST开始，变量说明必须以VAR开始。

语句部分必须以BEGIN开始，以END.结束，特别注意其中的小圆点（.）不可缺少。中间是一些用分号分开的语句。

一个PASCAL程序由程序首部和分程序组成，分程序通常由说明部分和语句部分组成（在特殊情况下可以没有说明部分）。

|  |
| --- |
| PROGRAM circle(input,output);  CONSTPI=3.14159;  VARr,l,s:real;  BEGINread(r);l:=2\*PI\*r;s:=PI\*r\*r;write(r,l,s);  END. |

|  |
| --- |
| 程序首部 |

|  |
| --- |
| 分程序 |

|  |
| --- |
| 说明部分 |

|  |
| --- |
| 语句部分 |

上例的第二行，大括号{ }中的内容，称为注释。被小括号及星号（\* \*）括起来的内容也是注释。注释除了给人看，以增加程序的可读性外，对编译和运行都不起作用，一个程序可以包含多个出现在不同地方的注释，也可以没有注释。

程序的书写格式：

1、PROGRAM写在最左边顶格；

2、注释的大括号{、和CONST、VAR、BEGIN、END等语句上下对齐，且它们比PROGRAM向右移两个字符；

3、各个语句和程序语句也是上下对齐，它们比第2点中的各语句又向右移两个字符；

4、使用大写字母的语句：PROGRAM、CONST、VAR、BEGIN、END；

5、使用大写字母的标识符：常量标识符。

**例 L0\_001** 输入三角形底边和高，求该三角形的面积。写出程序。

第二课  PASCAL的数据类型、标识符、表达式、语句

PASCAL的数据类型

数据对于一门程序语言是非常重要的，数据的一个非常重要的特征就是它的类型。PASCAL规定：程序中出现的变量必须先说明才能使用。

PASCAL的标准数据类型

|  |  |  |  |
| --- | --- | --- | --- |
| 数据类型 | 类型标识符 | 适用的数据 | 举例 |
| 实型 | real | 有小数表示法和指数表示法（科学表示法） | 1.25，132.67，0.0025，-1.561.3267e+2（=1.3267×102=132.67） |
| 整型 | integer | 包括正、负整数和零 | 25，-32，0 |
| 字符型 | char | 单个的字符 | ‘A’，’b’，’C’，’0’，’4’，’+’ |
| 布尔型 | boolean | 只有现个值，即true（真）和false（假） |   |

1、实型（real）

实型是最常用的数据类型。

在PASCAL，实数有两种表示方法：小数表示法和指数表示法（或称科学表示法）。

1.25，132.67，0.0025，-1.56，0.0，100.0

以科学表示法表示这些数，可以写成

1.25e0，1.3267e+2，2.5e-3，-1.56e0，0e0，1e2

在科学表示法中，e后的数字代表10的幂。上列各数可以解释成：

1.25e0=1.25×100=1.25，1.3267e+2=1.3267×102=132.67，2.5e-3=2.5×10-3=0.0025，-1.56e0=-1.56×100=-1.56，0e0=0×100=0.0，1e2=1×102=100.0

2、整型（integer）

整型数包括正、负整数和零。例如：25，-32，0。在整数中不得包括不数点（.）。

3、字符型（char）

字符型数是括在两个单引号中的一个字符。例如：

‘A’，’b’，’C’，’0’，’4’，’+’，’\*’，……

‘’代表空字符

‘’’’代表字符’

‘abc’不是字符型数，因为单引号中只能有一个字符。

所有字符都有一个序号与之对应。如‘0’对应48，‘1’对应49，‘A’对应65，‘B’对应66。

与字符有关的几个函数

|  |  |  |
| --- | --- | --- |
| 函数 | 功能 | 举例 |
| ord() | 返回与字符对应的序号 | ord(‘A’)=65 |
| chr() | 返回与数字对应的字符 | chr(65)=’A’ |
| pred() | 返回前一字符 | pred(‘B’)=A |
| succ() | 返回后一字符 | succ(‘A’)=B |

用函数可取得对应的序号，用

4、布尔型（boolean）

布尔类型的值只有两个：false（假）和true（真）。且false的序号为0，true的序号为1。

用于布尔量的运算有布尔运算（或称逻辑运算），它们是：

AND（与），OR（或），NOT（非），基本运算规则如下：

|  |  |
| --- | --- |
| 运算 | 结果 |
| 0 AND 0 | 假 AND 假 | 0 | false | 假 |
| 0 AND 1 | 假 AND 真 | 0 | false | 假 |
| 1 AND 0 | 真 AND 假 | 0 | false | 假 |
| 1 AND 1 | 真 AND 真 | 1 | true | 真 |
|   |   |   |   |   |
| 0 OR 0 | 假 OR 假 | 0 | false | 假 |
| 0 OR 1 | 假 OR 真 | 1 | true | 真 |
| 1 OR 0 | 真 OR 假 | 1 | true | 真 |
| 1 OR 1 | 真 OR 真 | 1 | true | 真 |
|   |   |   |   |   |
| NOT 0 | NOT 假 | 1 | true | 真 |
| NOT 1 | NOT 真 | 0 | false | 假 |

**例L0-002**  写出下列PASCAL语句的执行结果

   A0:=(1>5);

   A1:=(1<5);

   writeln(A0 AND A0);

   writeln(A0 AND A1);

   writeln(A1 AND A0);

   writeln(A1 AND A1);

   writeln(A0 or A0);

   writeln(A0 or A1);

   writeln(A1 or A0);

   writeln(A1 or A1);

标识符

标识符是以字母开头的字母、数字组合，用来表示常量、变量、类型、文件、函数、过程或程序的名字。

x，y，max，min，sum，a15，a3b7都是合法的标识符。而

5x，x-y，α，π，ε，ex10.5都是非法的标识符。

标识符的长度一般不要超过8个字符。

标识符的选取最好有一定意义，这样便于记忆，也增加了程序的可读性。

表达式和运算符

运算符

+ ， - ， \* ， /

AND，OR，NOT

，，，，，，，，，，，，，，，，，，，

表达式就是将数据和运算符结合使用，组成一组有意义的运算式。

在PASCAL语言中规定，表达式按下列运算优先规则计算：

1、所有括起来的子表达式必须先计算，且子表达式必须从里到外计算；

2、在同一子表达式中的运算符按下列次序计算：

①函数；②NOT；③AND，\*，/，DIV，MOD；④OR，+，-；⑤<，<=，=，>，>=，<>

3、在同一个子表达式中，同一优先级的运算按从左到右的次序进行。

下面是几个PASCAL表达式的例子：

**例L0\_003**

|  |  |
| --- | --- |
| 数学表达式 | PASCAL表达式 |
| b2-4ac | b\*b-4\*a\*c |
|   | (a+b)/(c+d) |
|   | (a+b)/c+d |
|   | a\*b/(c\*d)或a\*b/c/d |
|   |   |
| a2+bx+c |   |
|   |   |
|   |   |

常用的函数

|  |  |  |
| --- | --- | --- |
| 词汇 | 功能 |   |
| abs | 绝对值 |   |
| sqr | 平方 |   |
| sqrt | 开方 |   |
| ord | 取序号 | 参数为字符型变量，结果为该字符的序号（整型） |
| chr | 取字符 | 参数为整型变量，结果为以参数为序号的字符 |
| pred | 前导字符 |   |
| succ | 后继字符 |   |
| odd | 奇函数，判断是否为奇数 | 参数为整型变量，结果为布尔型 |
| trunc | 去掉小数部分，取其整数 |
| round | 四舍五入，取其整数 |
| exp | 以e为底的指数 |
| ln | 以e为底的自然对数 |

函数的格式

例：开方函数， 在PASCAL中可表示为sqrt(2)

语句

赋值语句（:=）

读语句（read，readln）

当实型数据和字符型数据一起读入时，要先读字符型数据。

例如a、b、c为实型变量，ch1、ch2、ch3为字符型变量，可以写如下语句：

read(ch1,ch2,ch3,a,b,c)

不能写成：

read(a,b,c,ch1,ch2,ch3)

写语句（write，writeln）

write语句是写语句（或称输出语句）。用于将计算结果通过屏幕或打字机输出给人看。

例如：

write(sum,ave)

将在屏幕或打字机上输出sum和ave的值。

它的一般形式是

write(<输出表>)

其中输出表是一些由逗号分开的输出项。例如：

write(x,y,x+y,x\*y)

若x的值为2，y的值为3（整数类型），则输出结果为：

2   3   5    6

为了标记输出结果，可输出相应的字符串。例如将上面的语句改为：

write(‘x=’,x,’y=’,y,’x+y=’,x+y,’x\*y=’,x\*y)

则输出结果为：

x=2  y=3 x+y=5  x\*y=6

显然，现在的输出结果，较前面清楚。

为了将输出结果打印在不同的行上，可以利用writeln语句。它的作用是在印出输出表的最后一个输出项后，结束当前输出行，使得下一个write（或writeln）语句从下一个新行的开头（左端）输出。例如：

writeln(‘x=’,x,’y=’,y);

writeln(x+y=’,x+y,’x\*y=’,x\*y)

输出结果为：

x=2  y=3

x+y=5  x\*y=6

write语句可以输出实型、整型、字符型、布尔型值，也可以输出字符串。若在write语句中不加场宽说明，则按标准场宽输出。在这里，场宽是指输出值的位数。

|  |  |
| --- | --- |
| 数据类型 | 场宽 |
| 实型 | 以科学表示法输出，占16位 |
| 整型 | 12位 |
| 字符型 | 1位 |
| 布尔型 | 输出true或false，分别占5位和4位 |

当输出值不满规定场宽时，左面用空格代替。不同计算机系统的标准场宽规定可能不同。

-1.256734121e+02      13257Afalse

|  |
| --- |
| 实型16位 |

|  |
| --- |
| 整型12位 |

|  |
| --- |
| 字符型1位 |

|  |
| --- |
| 布尔型5位 |

**例L2\_2**  已知三角形的两边及夹角，求第三边及面积。

      设三角形的两边及夹角分别为a，b，α，第三边为c，面积为s。

则  ，

若α以角度值输入，在计算sin和cos时应转换为弧度。角度转弧度的公式为：

弧度=角度×

PROGRAM XT2\_2;

  CONST

   PI=3.14159;

  VAR

   a,b,alfa,c,s:real;

  BEGIN

   read(a,b,alfa);

   alfa:=alfa\*PI/180;

   c:=sqrt(a\*a+b\*b-2\*a\*b\*cos(alfa));

   s:=1/2\*a\*b\*sin(alfa);

   write(alfa,c,s);

  END.

**例L2\_3**输入一个字符，求其序号、前导（即前一字符）、后继（即后一字符）。

PROGRAM L2\_3;

  VAR

   ch,pch,sch:char;

   num:integer;

  BEGIN

   writeln;

   writeln;

   write('Please input a character:');

   readln(ch);

   write('pch:',pred(ch),',sch:',succ(ch),',num:',ord(ch))

  END.

**例L2\_4** 输入x，y。若在圆环内，输出true，若在圆环外，；输出false。圆环如图所示。

讨论：如图所示，若下式

1≤x2+y2≤4

成立则在圆环内，否则在圆环外。

设布尔变量bool，当x，y在圆环内时，让它取值为true，否则取值为false。

|  |
| --- |
| x |

|  |
| --- |
| y |

|  |
| --- |
| 1 |

|  |
| --- |
| 2 |

|  |
| --- |
| 0 |

“（x，y）若在圆环内”的表达式：(x2+y2≥12)且(x2+y2≤22)

PROGRAM L2\_4;

  VAR

   x,y:real;

   bool:boolean;

  BEGIN

   writeln;

   write('x=?');

   readln(x);

   write('y=?');

   readln(y);

   bool:=(x\*x+y\*y>=1) and (x\*x+y\*y<=4);

   write(bool);

  END.

**习题**

**XT1\_1**  判断下列标识符，哪些是合法的？哪些是非法的？

x3，3x，a17，p5q，π，β，ε，abcd，x2，ex9.5

**XT1\_2** 输入三个数，计算并输出它们的平均值以及三个数的乘积，写出程序。

**XT1\_3** 已知地球半径为6371km，计算并输出地球的表面积和体积，写出程序。球表面积计算公式s=4πR2 ，球体积计算公式 （其中R指球半径）。

**XT1\_4** 已知匀加速运动的初速度为10m/s，加速度为2m/s2，求20s以后的速度，20s内走过的路程及平均速度，写出程序。

设初速度为v0，加速度为a，时间为t，末速度为vt，路程为s，平均速度为vave，求末速度公式：vt=v0+at，路程的公式： ，平均速度为vave=

**XT1\_5**  读入摄氏温度c，将它转换成华氏温度 输出，写出程序。转换公式：

**XT2\_1**  下列数据哪些是整数？哪些是实数？哪些是非法的数？

256，2.50，1e+06，2.2e5，e10，-785，e-5，

.5，15.，15.0，0.12，0，0.0，25e，

**XT2\_2** 确定下列各常数、表达式的类型和值，表达式类型是指表达式值（结果）的类型。

‘A’，’3’，3，true，false，

12/3，12 DIV 3，5>3，5<3，

105 MOD 10，round(17/3，trunc(17/3)，

sqrt(ord(‘2’)-ord(‘0’))，

sqr(ord(‘2’)-ord(‘0’))，

(5>3) AND (5<3)，chr(67)

**XT2\_3**  将下列数学表达式表示成PASCAL表达式

(1)-(a2+b2)·y4   (2)sin2(x+0.5)+3cos(2x+4)+tan-1x   (3)u0(1+e-0.5t)

(4)     (5)     (6)

(7)    (8)|3-exln(1+x)|   (9)     (10)

**XT2\_4** 写一程序读入三角形的三个边a，b，c，计算并打印三角形的面积S。可利用下列公式计算。

其中

**XT2\_5**  按下列公式计算和打印y值

|  |
| --- |
| r |

|  |
| --- |
| h1 |

|  |
| --- |
| h2 |

|  |
| --- |
| XT2\_7图 |

**XT2\_6** 输入二次方程的系数a，b，c（输入数据的选取应保证二次方程为实根），计算并输出二次方程的两个实根。二次方程为：

ax2+bx+c=0

**XT2\_7** 求图形的表面积与体积。其中r，h1，h2的值由键盘读入。，，

**XT2\_8** 输入三个字符，然后按输入字符次序输出这三个字符，再输出每个字符的序号，最后按与输入字符相反的次序输出这三个字符。

**XT2\_9** 由键盘输入两组x，y值，由程序根据它们是否在示意图中的斜线区域内，输出不同的值。若在斜线区域内，输出true，否则输出false。

**XT2\_10** 输入一个三位整数，将它们反向输出。例如输入127，输出应为721。

**XT2\_11** 输出你的中文（或英文）名字。每个汉字（或字母用7行5列字符描述。例如王平输出如图所示。

|  |
| --- |
| x |

|  |
| --- |
| y |

|  |
| --- |
| 2 |

|  |
| --- |
| 2 |

|  |
| --- |
| -2 |

|  |
| --- |
| -2 |

|  |
| --- |
| 0 |

|  |
| --- |
| XT2\_9图 |

|  |
| --- |
| \*\*\*\*\*  \*  \*\*\*\*\*\*  \*  \*\*\*\*\*\* |

|  |
| --- |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*  \*\* |

|  |
| --- |
| XT2\_11图  |

第3课  选择结构程序设计（20071224）

|  |
| --- |
| **请参阅《全国信息学奥林匹克联赛培训教程（一）》P101****5.3.3** **选择结构** |

**3.1** **引言**

通常，计算机按在程序中书写的顺序执行语句。然而，在许多情况下，语句执行的顺序依赖于输入数据或中间运算结果。在这种情况下，必须根据某个变量或表达式的值作出判定，以决定执行哪些语句和跳过哪些语句不执行。我们称这种程序结构为选择结构（或称判定结构）。

PASCAL中有两个语句可实现选择结构：IF语句（条件语句）和CASE语句（情况语句）

**3.2  IF语句**

**3.2.1  IF语句的两种形式**

|  |
| --- |
| 条件 |

|  |
| --- |
| 语句2 |

|  |
| --- |
| 语句1 |

|  |
| --- |
| 假 |

|  |
| --- |
| 真 |

IF语句又称如果语句（或称条件语句）。它的一般形式是

IF （条件）

  THEN （语句1）

  ELSE （语句2）

其中“条件”实际上是一个布尔表达式，它的值可以是真（true）或假（false）。在条件为真时，执行语句1，否则（条件为假）执行语句2。

可以利用流程图来描述选择结构。

在流程图中用菱形框表示要执行的判定，其余步骤用矩形框表示，箭头表示执行的顺序。

**例L3\_1** 火车托运行李，要根据行李的重量按不同标准收费。例如不超过50kg，按每公斤0.35元收费。若超过50kg，则其中50kg按每公斤0.35元收费，其余超过部分按每公斤0.50元收费。现输入托运行李重量，要求计算并输出托运费。

解：

讨论：设托运行李重量为weight，该问题要求根据weight是否超过50kg，按不同公式计算出托运费。设托运费为pay。weight和pay均为实型。

下面给出算法。

一级算法

1、读入weight的值。

2、根据weight的大小选择不同的公式计算pay。

3、输出pay。

其中第2步需求精。根据题意，可以写出计算pay的公式如下：

|  |
| --- |
| weight≤50weight>50 |

二级求精

|  |
| --- |
| weight≤50 |

|  |
| --- |
| pay=50×0.35+(weight-50) ×0.5 |

|  |
| --- |
| pay=weight×0.35 |

|  |
| --- |
| 假 |

|  |
| --- |
| 真 |

|  |
| --- |
| L3\_1图 |

第2步 计算pay

2-1  如果weight≤50

2-2  则pay=weight×0.35

2-3  否则pay=50×0.35+（weight-50）×0.5

以上求精步骤要求首先判断weight是否小于等于50。如果小于等于50应执行算法步骤2-2，否则应执行算法步骤2-3。即算法步骤2-2和2-3不是同时执行的，需根据情况选择其一执行。

PROGRAM L3\_1;

  VAR

   weight,pay:real;

  BEGIN

   write('weight=');

   readln(weight);

    IF weight<=50

     THEN pay:=0.35\*weight

     ELSE pay:=0.35\*50+0.5\*(weight-50);

   writeln('pay=',pay)

  END.

除了上面给出的IF语句形式外，PASCAL中还有另外一种IF语句形式。即

|  |
| --- |
| 条件 |

|  |
| --- |
| 语句1 |

|  |
| --- |
| 假 |

|  |
| --- |
| 真 |

IF （条件）

  THEN （语句）

在条件为真时，执行THEN后的语句。在条件为假时，不执行THEN后的语句，在两种情况下的后继语句都是IF语句的下一个语句。

**例L3\_2**  读入三个数，找出并打印其中的最大数。

解：

PROGRAM L3\_2;

  VAR

   a,b,c:real;

  BEGIN

   write('a=?');

   read(a);

   write('b=?');

   read(b);

   write('c=?');

   read(c);

    IF a<b

     THEN a:=b;

    IF a<c

     THEN a:=c;

   writeln('ZuiDaShu:',a)

  END.

**3.2.2** **复合语句**

在IF语句中，跟在THEN或ELSE后的语句可能不止一个，这时要用到复合语句的概念。

复合语句是一个以BEGIN开始，以END结束的语句。在BEGIN与END之间可以包括若干个语句，每个语句之间以分号分开。一般形式为：

BEGIN

  （语句1）；

  （语句2）；

  …

  （语句n）

END

一个复合语句从外部看来，相当于一个语句。

**例L3\_3**  读两个数，将大数存于x，小数存于y。

解：

PROGRAM L3\_3;

  VAR

   x,y,temp:real;

  BEGIN

   write('x=?');

   readln(x);

   write('y=?');

   readln(y);

    IF x<y

     THEN BEGIN

            temp:=x;

            x:=y;

            y:=temp

          END;

   writeln('x=',x:8:2,',y=',y:8:2)

  END.

**3.2.3** **复合IF语句**

在IF语句中，THEN或ELSE后的语句本身也可能是IF语句。此时称为IF语句的嵌套（或称为复合IF语句）。

|  |
| --- |
| 条件1 |

|  |
| --- |
| 语句1 |

|  |
| --- |
| 假 |

|  |
| --- |
| 真 |

|  |
| --- |
| 条件2 |

|  |
| --- |
| 假 |

|  |
| --- |
| 真 |

|  |
| --- |
| 语句2 |

|  |
| --- |
| 语句3 |

例如语句

IF（条件1）

  THEN（语句1）

  ELSE IF（条件2）

         THEN（语句2）

         ELSE（语句3）

就是一个复合IF语句，在它的ELSE后又是一个IF语句。

**例L3\_4** 输入某学生成绩，根据成绩的好坏输出相应评语。如果成绩在90分以上，输出评语：优秀（outstanding）；如果成绩在60到90之间，输出评语：良好（satisfactory）；如果成绩不足60分，输出评语：不及格（unsatisfactory）。

有时IF语句可能会有两种不同的理解。例如语句：

IF （条件1）

  THEN IF （条件2）

          THEN （语句1）

          ELSE （语句2）

我们可以认为ELSE与后一个THEN配对，其流程图如图3\_8(1)所示。

但是，我们也可以认为ELSE与前一个THEN配对，其流程图如图3\_8(2)所示。

|  |
| --- |
| 条件1 |

|  |
| --- |
| 语句1 |

|  |
| --- |
| 假 |

|  |
| --- |
| 真 |

|  |
| --- |
| 条件2 |

|  |
| --- |
| 假 |

|  |
| --- |
| 真 |

|  |
| --- |
| 语句2 |

|  |
| --- |
| 图3\_8(1) |

|  |
| --- |
| 条件1 |

|  |
| --- |
| 语句1 |

|  |
| --- |
| 假 |

|  |
| --- |
| 真 |

|  |
| --- |
| 条件2 |

|  |
| --- |
| 假 |

|  |
| --- |
| 真 |

|  |
| --- |
| 语句2 |

|  |
| --- |
| 图3\_8(2) |

为了避免产生不同的理解，PASCAL系统规定ELSE总是与同一层前面最靠近它的，还没有ELSE与之配对的THEN配对。因此，上面的语句肯定是按第一种理解，即ELSE与后一个THEN配对。

如果要求按第二种理解执行流程图，可以通过在IF语句中加BEGIN、END的办法来实现。写成

IF（条件1）

THEN BEGIN

       IF（条件2）

       THEN（语句1）

     END

ELSE（语句2）

**3.3  CASE语句**

CASE语句是实现选择结构程序设计的另一种语句。它的使用有时比IF语句来得简单、直观。

CASE语句（或称情况语句）的一般形式是

CASE （表达式） OF

  （值表1）：（语句1）；

  （值表2）：（语句2）；

    ……

  （值表n）：（语句n）

END

在CASE语句头上的表达式必须是有序类型（整型、字符型、布尔型以及后面要介绍的枚举型、子界型）。值表是一些由逗号分开的常数。表达式所有可能的值必须在值表中出现，且每个值只能出现一次。

如果当前表达式的值在某个值表i中出现，则该程序只执行对应该值表i的语句，然后执行整个CASE语句后的下一语句。

**例L3\_5**  输入年、月，输出该月有几天。

讨论：每年的1、3、5、7、8、10、12月，每月有31天；4、6、9、11月，每月有30天；2月闰年有29天，平年有28天。

年号能被4整除，但不能被100整除，或者年号能被400整除的年均是闰年。

用year、month、days分别表示年、月、每月天数。它们均为整数。

闰年的条件可以写成如下的布尔表达式：

(year MOD 4 = 0) AND (year MOD 100 <> 0) OR (year MOD 400 = 0)

PROGRAM L3\_5;

  VAR

   year,month,days:integer;

  BEGIN

   write(‘year=?’);

readln(year);

   write(‘month=?’);

readln(month);

CASE month OF

  1,3,5,7,8,10,12:days:=31;

  4,6,9,11:day:=30;

     2:IF(year MOD 4 = 0) AND (year MOD 100 <> 0) OR (year MOD 400 = 0)

      THEN days:=29

      ELSE days:=28

    END;

    writeln(‘year=’,year,’,month=’,month,’,days=’days)

   END.

**例L3\_6** 输入两个运算量及一个运算符，输出运算结果。这相当于计算器计算。

讨论：设两个运算量为x和y，一个运算符为oper，运算结果为result。

现在假定运算符只可能是+、-、\*、/，将oper说明为字符型。x、y和result说明为实型。

**习题**

**XT3\_1**  编制程序，根据输入的x值，计算y与z并输出。

|  |
| --- |
| PROGRAM XT3\_1;  CONST   PI=3.14159;  VAR   x,y,z:real;  BEGIN   write('x=?');   read(x);    IF x<=2.5     THEN y:=x\*x+1     ELSE y:=x\*x-1;    IF x<0     THEN z:=-PI/2\*x+3     ELSE IF x=0            THEN z:=0            ELSE z:=PI/2\*x-5;   writeln('x=',x:6:2,', y=',y:6:2,', z=',z:6:2)  END. |

**XT3\_2** 对一批货物征收税金，价格在1万元以上的货物征税5%，在5000元以上，1万元以下的货物征税3%，在1000元以上，5000元以下的货物征税2%，1000元以下的货物免税。编写一程序，读入货物价格，计算并输出税金。

**XT3\_3**  输入某学生成绩，若成绩在85分以上，输出very good，若成绩在60分到85分之间，输出good，若成绩低于60分，输出no good。

**XT3\_4**  输入3个数，将其中最小数输出。

**XT3\_5**  输入3个字母，按字母表顺序从小到大输出这3个字母。

**XT3\_6** 输入二次方程的3个系数a、b、c，计算二次方程的根。程序应考虑实根和复根两种情形。二次方程如下：

ax2+bx+c=0

|  |
| --- |
| z= |

|  |
| --- |
| lnx+lnysinx+sinye2x+e3xtan(x+y) |

|  |
| --- |
| Ⅰ象限Ⅱ象限Ⅲ象限Ⅳ象限 |

提示：PASCAL不能表示复数。若为复根，应分别计算并输出实、虚部，在输出时将+、-与i用字符形式输出。

**XT3\_7**  输入x、y，根据所处象限，用相应公式计算z。

**XT3\_8** 输入x、y及象限，计算z。z的计算公式同XT3\_7题，要求用CASE语句编程序。

**XT3\_9**  输入班号，输入该班学生人数。应用CASE语句编程序。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 班号 | 91 | 92 | 93 | 94 | 95 | 96 |
| 人数 | 30 | 32 | 30 | 30 | 32 | 25 |

阅读学习《全国信息奥林匹克联赛培训教程（一）》（信息学奥林匹克科学委员会主编，清华大学出版社）及网上教案，阅读要求见下表：

|  |  |  |
| --- | --- | --- |
| 阅读内容 | 页  码 | 要求 |
| 5.2  Pascal程序的一般结构和输入输出语句 | 《全国信息奥林匹克联赛培训教程（一）》P94 | 全看 |
| 5.3.2  顺序结构 | 《全国信息奥林匹克联赛培训教程（一）》P100 | 全看 |
| 5.3.2  选择结构 | 《全国信息奥林匹克联赛培训教程（一）》P101 | 看1、条件选择结构P101至P104 |
| 5.4.1  运算对象的显式定义 | 《全国信息奥林匹克联赛培训教程（一）》P116 | 全看 |
| 5.4.2  表达式 | 《全国信息奥林匹克联赛培训教程（一）》P120 | 全看 |
| 5.4.3  Pascal的简单类型 | 《全国信息奥林匹克联赛培训教程（一）》P121 | 看1、整型；2、布尔类型；4、字符类型；6、实数类型 |
| CHH信息学奥赛培训教案（20071127） | http://edu.jianlan.com.cn/TuoZhan/NOI/kwtz\_NOIP.htm | 全看 |

阅读后完成下面题目

1、Pascal程序可以分为三个部分，请将这三个部分的名称填入下面括号中

|  |
| --- |
| PROGRAM circle(input,output);  CONSTPI=3.14159;  VARr,l,s:real;  BEGINread(r);l:=2\*PI\*r;s:=PI\*r\*r;write(r,l,s);  END. |

|  |
| --- |
| （         ） |

|  |
| --- |
| （         ） |

|  |
| --- |
| （         ） |

2、完成下面有关数据类型的表

|  |  |  |
| --- | --- | --- |
| 数据类型 | 类型标识符 | 举2个例子 |
| 实型 |   |   |
|   | integer |   |
| 字符型 |   |   |
|   | boolean |   |

3、请将下列实型数据用另一种表示法表示

|  |  |  |  |
| --- | --- | --- | --- |
| 小数表示法 | 科学表示法 | 小数表示法 | 科学表示法 |
| 例：12.5 | 1.25e+1 |   | 1e+2 |
| 132.67 |   |   | -3.05e+1 |
| 0.0025 |   |   | -2.509e-4 |
| -1.56 |   |   | 5.201e-2 |
| 5982.37 |   |   | 3.8e+2 |
| 0.84 |   |   | 3.7e-3 |
| -0.0071 |   |   | 3.05e0 |

4、在Pascal中，

有六个算术运算符，它们是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

有六个关系运算符，它们是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

有三个逻辑运算符，它们是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

5、在Pascal中，各种运算符的运算优先规则如下，请在下面直线上画一个箭头表示运算优先级由低到高并在直线上面的括号内填入优先级的高低。

|  |
| --- |
| 优先级（    ） |

|  |
| --- |
| 括号（） |

|  |
| --- |
| 函数 |

|  |
| --- |
| NOT |

|  |
| --- |
| AND\*/DIVMOD |

|  |
| --- |
| OR+－ |

|  |
| --- |
| < <==>=> <>  |

|  |
| --- |
| 优先级（    ） |

6、填出下面Pascal词汇所属类别（在“语句、函数、运算符、数据类型、符号”中选择其一）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **序号** | **词汇** | **类别** | **功能** | **序号** | **词汇** | **类别** | **功能** |
| **1** | := |   |   | **30** | sqr |   | 平方 |
| **2** | abs |   | 绝对值 | **31** | sqrt |   | 开方 |
| **3** | AND |   | 逻辑运算符“与” | **32** | succ |   | 后继字符 |
| **4** | BEGIN |   | 分程序开始 | **33** | THEN |   | 条件语句IF…THEN…ELSE的一部分 |
| **5** | boolean |   |   | **34** | TO |   | FOR循环语句的一部分 |
| **6** | CASE |   | 分支结构语句 | **35** | trunc |   | 去掉小数部分，取其整数 |
| **7** | char |   |   | **36** | VAR |   | 说明变量 |
| **8** | chr |   | 取字符 | **37** | WHILE |   | 循环语句WHILE |
| **9** | CONST |   | 定义常量 | **38** | sin |   |   |
| **10** | DIV |   | 整数商 | **39** | cos |   |   |
| **11** | DO |   | WHILE DO语句的一部分 | **40** | arctan |   |   |
| **12** | ELSE |   | 条件语句IF…THEN…ELSE的一部分 | **41** | + |   |   |
| **13** | END |   | 结束分程序 | **42** | - |   |   |
| **14** | exp |   | 以e为底的指数 | **43** | \* |   |   |
| **15** | FOR |   | FOR循环语句 | **44** | / |   |   |
| **16** | GOTO |   |   | **45** | <  |   |   |
| **17** | IF |   | IF条件语句 | **46** | <= |   |   |
| **18** | integer |   |   | **47** | = |   |   |
| **19** | ln |   | 以e为底的自然对数 | **48** | >= |   |   |
| **20** | MOD |   | 取余，整除后的余数 | **49** | >  |   |   |
| **21** | NOT |   | 逻辑运算符“非” | **50** | <>  |   |   |
| **22** | odd |   | 奇函数，判断是否为奇数 | **51** | write |   |   |
| **23** | OF |   | 分支结构语句CASE…OF的一部分 | **52** | writeln |   |   |
| **24** | OR |   | 逻辑运算符“或” | **53** | read |   |   |
| **25** | ord |   | 取序号 | **54** | readln |   |   |
| **26** | pred |   | 前导字符 | **55** | : |   |   |
| **27** | PROGRAM |   | 程序首部 | **56** | ' |   |   |
| **28** | real |   |   | **57** | ; |   |   |
| **29** | round |   | 四舍五入，取其整数 |  |   |   |   |

2、在第1题例中，语句write与writeln有何不同\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

3、在第1题例中，哪些词属于保留字\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

5、将下列代数式写成Pascal的表达式

（1）4×[（13-4）÷6]       （2）a-(m-d)×4÷b+3y=A2

（3）            （4）

6、下列符号中，能作标识符的是：

AI, Wang, 4D4g, IF, MAX, ABS, PI, SIN, ABS\_1, Begin

7、写出以下表达式的值及类型

(1)10/2+2

重复控制结构（循环结构）

循环结构（或称重复结构）是程序中的一个基本结构，在解许多问题中是很有用的。我们知道，在许多复杂的问题中，常常需要做大量类同的计算处理。尽管计算机的运算速度很快，然而要把这些大量类同的计算处理的每一步都写成语句，并输入计算机中，其工作量是相当大的。有时是难以完成的。

循环结构程序设计可以帮助我们有效地解决这一难题。利用循环结构程序设计，使得我们有可能只编写少量的语句，让计算机重复执行它许多次，从而完成大量类同的计算要求。

在PASCAL中，实现循环程序设计的主要语句有FOR语句、WHILE语句和REPEAT语句。

一、FOR语句

FOR x：=<初值> TO <终值> DO

  <语句>

FOR  X:= 1 TO  10  DO

FOR  X:=10 DOWNTO  1  DO

FOR  X:= ‘A’ TO  ‘Z’  DO

FOR  X:=’Z’ DOWNTO  ‘A’  DO

|  |
| --- |
| s=0当n=1   s=s+n=0+1=1;当n=2   s=s+n=1+2=3当n=3   s=s+n=3+3=6当n=4   s=s+n=6+4=10当n=5   s=s+n=10+5=15当n=6   s=s+n=15+6=21当n=7   s=s+n=21+7=28当n=8   s=s+n=28+8=36当n=9   s=s+n=36+9=45当n=10   s=s+n=45+10=55  |

T75\_001  计算1+2+3+…+10

PROGRAM T75\_001;

  VAR

   s,n:integer;

  BEGIN

    s:=0;

    FOR n:=1 TO 10 DO

     s:=s+n;

   writeln(‘s=’,s)

  END.

T75\_012  计算n!（n!=1×2×3×…×n）

T75\_013  输入10个任意整数，求它们的和

PROGRAM T75\_001;

  VAR

   s,n,a:integer;

  BEGIN

    s:=0;

    FOR n:=1 TO 10 DO

     BEGIN

       read(a);

       s:=s+a;

     END;

   writeln(‘s=’,s)

  END.

T75\_014  输入10个任意整数，求它们的平均值

T75\_015  按正序和反序输出26个英文字母

T75\_016  输入20个整数，输出其中最大数

T75\_017  输入20个整数，统计其中正、负和零的个数

T75\_001,T75\_012~T75\_017共七题的答案：

|  |
| --- |
| PROGRAM T75\_012;  VAR   n,i,sum:integer;  BEGIN   read(n);   sum:=1;    FOR i:=1 TO n DO     sum:=sum\*i;   writeln('n!=',sum);  END. |

|  |
| --- |
| PROGRAM T75\_001;  VAR   s,n:integer;  BEGIN    s:=0;    FOR n:=1 TO 10 DO       s:=s+n;   writeln('s=',s)  END. |

|  |
| --- |
| PROGRAM T75\_014;  VAR   ave:real;   i,n,sum:integer;  BEGIN   sum:=0;    FOR i:=1 TO 10 DO     BEGIN       read(n);       sum:=sum+n;     END;   ave:=sum/10;   writeln('ave=',ave:6:2)  END. |

|  |
| --- |
| PROGRAM T75\_013;  VAR   s,n,a:integer;  BEGIN    s:=0;    FOR n:=1 TO 10 DO     BEGIN       read(a);       s:=s+a;     END;   writeln('s=',s)  END. |

|  |
| --- |
| PROGRAM T75\_015;  VAR   i:integer;  BEGIN   writeln();    FOR i:=1 TO 26 DO     write(chr(96+i));   writeln();    FOR i:=122 DOWNTO 97 DO     write(chr(i))  END. |

|  |
| --- |
| PROGRAM T75\_016;  VAR   maximal,i,n:integer;  BEGIN   read(maximal);    FOR i:=1 TO 19 DO     BEGIN       read(n);       IF maximal<n         THEN maximal:=n;     END;   writeln('max=',maximal)  END. |

|  |
| --- |
| PROGRAM T75\_017;  VAR   i,n,Zheng,Fu,Ling:integer;  BEGIN   Zheng:=0;   Fu:=0;   Ling:=0;    FOR i:=1 TO 20 DO     BEGIN       read(n);       IF n>0         THEN Zheng:=Zheng+1         ELSE IF n=0                THEN Ling:=Ling+1                ELSE Fu:=Fu+1;     END;{FOR}   writeln('Zheng Shu You: ',Zheng,'; Fu Shu You: ',Fu,'; Ling You: ',Ling)  END. |

二、WHILE语句

当循环次数未知，例如它依赖于某个布尔表达式的值，而此值在循环执行过程中会改变。这种循环不可能由FOR语句实现，PASCAL为此提供了WHILE语句和REPEAT语句。

WHILE语句的一般形式：

WHILE <布尔表达式>  DO

<循环体>

T75\_002  输入一串字符，以‘？’结束，输出其中字母个数与数字个数。

执行该语句时，首先检查布尔表达式的值，如果它为真，则执行循环体。在循环体中通常包括改变布尔表达式值的语句。每次执行循环体后，再次检查布尔表达式，如果它仍为真，继续执行循环体，否则循环结束，执行WHILE语句后的下一语句。当一开始布尔表达式就为假时，循环体根本不执行。

在WHILE语句中一定要有能改变布尔表达式值的语句，否则循环将不会结束。

三、REPEAT语句

REPEAT语句也用于循环次数未知的循环，它的用法与WHILE语句稍有不同。

REPEAT语句的一般形式如下：

REPEAT

<循环体>

UNTIL <布尔表达式>

T75\_003  计算    直到最后一项的绝对值小于10-7时停止计算，x由键盘输入。

sin(x)本来可以利用系统提供的标准函数直接写出。但是标准函数是如何计算sin(x)的呢？通过这个例子，将使我们对此有所了解，并可作为使用REPEAT语句的例子。

T43\_004  以下三种IF的写法都是错误的，请指出错在哪里。

…

ch:char;

…

read(ch);

IF (ch>=a) AND (ch<=z)

…

IF ch>=’a’ AND ch<=’z’

…

IF ‘a’<=ch<=’z’

…

三种循环语句的相同点与不同点

|  |  |  |  |
| --- | --- | --- | --- |
|   | FOR | WHILE | REPEAT |
| 语句格式 | FOR <循环变量>：=<初值> TO <终值> DO  <语句>  | WHILE <布尔表达式>  DO<循环体>  | REPEAT<循环体>UNTIL <布尔表达式>  |
| 循环变量 | 赋值 | 布尔表达式 | 无循环变量 |
| 循环次数 | **确定**，由初值和终值决定 | **不确定**，由循环体前的布尔表达式决定，当值为“假”时，结束循环。可能一次循环也没有 | **不确定**，由循环体后的布尔表达式决定，当值为“真”时结束循环。至少有一次循环 |
| 循环体 | 多语句时，需用BEGIN和END | 多语句时，需用BEGIN和END | 多语句时，不需用BEGIN和END |
|   |   |   |   |
|   |   |   |   |
|   |   |   |   |

用计算机解题的基本方法

1、问题分析

计算机是用于表示和处理数据的工具。因此解题的前两个任务：一是定义在计算机存储器中表示的数据；二是描述算法——列出处理这些数据的步骤。这两个任务不是完全无关的。当描述算法时，可以改变数据定义。然而在构造算法前，应尽可能完全和仔细地招待数据定义。如果数据定义有错，描述算法是困难的，甚至是不可能的。

 定义数据要求清楚地了解问题。首先必须确定由计算机计算和打印的信息，然后标识输入给计算机的信息。一旦标识了输入输出，我们要问，为了从给定的输入计算输出，是否有足够的信息可利用？如果没有，必须确定附加的信息，以及提供这些信息的手段。

当标识与问题有关的数据项时，要为每一项赋一个助忆名。它可以用来代表存储数据项的计算机存储单元，而不必考虑与每个变量名相联系的实际存储单元。编译程序将为每个变量名赋一个唯一的存储单元。

2、问题解的描述

在对问题有了清楚的分析后，可以仔细地构造求解步骤——算法。算法可以自顶向下、由粗到细，逐步求精。

描述问题由粗到细的过程，一般可以分为三步：

一级算法

二级求精

写出程序

四、多重循环

如果一个循环结构的内部（循环体）又包括一个循环结构，就称为多重循环结构。实现多重循环结构仍可以用前面讲的三种循环语句。因为任一循环语句的循环体部分都可以包含另一个循环语句，这种循环语句的嵌套为实现多重循环提供了方便。

多重循环的嵌套次数可以是任意的。可以按照嵌套层次数，分别叫做二重循环、三重循环等。处于内部的循环叫作内循环，处于外部的循环叫作外循环。

在设计多重循环时，要特别注意内、外循环之间的关系，以及各语句安放的位置，不要搞错。

T75\_007  求1~100之间的素数(质数).

素数是大于1，且除了1和它本身外，不能被其它任何整数所整除的整数。

讨论：根据素数的定义可知

2、3、5、7、11、13、17等是素数。

1、4、6、8、9、10、12、14、15等不是素数。

为了判断某数i是否为素数，一个最简单的办法是用2、3、4、5、……、i-1这些数逐个去除i，看能否除尽。若被其中一个数除尽了，则i不是素数，否则（全都除不尽）i是素数。

当i较大时，用这种办法，除的次数太多。可以有许多改进办法，以减少除的次数，提高运行效率。下面介绍的是其中一种办法。

用2、3、4、…… 去除，如果都除不尽，则i是素数。

这是因为，如果小于等于 的数都除不尽i，则大于 的数也不能除尽i。

用反证法证明：假设有大于 的数j能除尽i，则它的商k必小于 ，且k能除尽i（其商为j）。这是与原命题（小于等于 的数都除不尽i）相矛盾的，假设不成立，原命题得到证明。

一级算法：

FOR i:=2 TO 100 DO

  BEGIN

    1、用2到 去除i，看看能否除尽

    2、IF 除不尽

        THEN 输出素数i

  END

算法需进一步求精。用变量yn来表示是否除尽。

一开始让yn=1，当有一个数能除尽时，让yn=0表示能除尽，这时i不是素数；若循环结束时，yn仍等于1，表示都除不尽，这时i是素数。

二级求精

    1、用2到 去除i，看看能否除尽

   1-1  yn:=1

   1-2  WHILE n<sqrt(i) DO

         BEGIN

           n:=n+1

           IF i MOD n=0

             THEN yn:=0

         END

为了每行输出5个数，还需增加一个变量counter来计数。

PROGRAM T75\_007;

  VAR

   i,n,yn,counter:integer;

  BEGIN

   writeln;

   counter:=0;

    FOR i:=2 TO 100 DO

     BEGIN

       n:=1;

       yn:=1;

       WHILE n<sqrt(i) DO

         BEGIN

           n:=n+1;

           IF i MOD n=0

             THEN yn:=0;

         END;{WHILE}

       IF yn<>0

         THEN BEGIN

                counter:=counter+1;

                IF counter MOD 5=0

                  THEN writeln(i)

                  ELSE write(i,' , ');

              END;{IF}

     END;{FOR}

  END.

T75\_008  哥德巴赫猜想

验证哥德巴赫猜想（任何充分大的偶数都可由两个素数之和表示）。将4~2\*s（s≤1000）中的所有偶数分别用两个素数之和表示。例如s=50，则输出

4=2+2

6=3+3

8=3+5

……

100=3+97

讨论：哥德巴赫猜想是一个古老而著名的数学难题。它的理论证明很麻烦，迄今未得出最后的证明。在这方面中国数学家陈景润的研究成果处于世界领先地位。

在陈景润以前的数学家只证明出：对任一充分大的偶数even，可以找到4个素数p、q、r、s。使得

even=p+q\*r\*s

这也称为1+3问题

陈景润证明出：对任一充分大的偶数even，可以找到3个素数p、q、r。使得

even=p+q\*r

这也称为1+2问题。

而真正的哥德巴赫猜想是：对任一充分大的偶数even，可以找到两个素数p、q。使得

even=p+q

即1+1问题。此问题还未得到最后的证明。

我们这里只是对有限范围内的数，用计算机加以验证，不算严格证明。

读入偶数even，将它分成p和q，使even=p+q。

p从2开始（每次加1），q=even-p。如果p、q均为素数，则输出结果，否则将p加1再试。

一级算法：

1、读入一个数even；

2、判断even是否为偶数，如果是偶数，则继续，否则结束程序；

3、FOR p:=2 TO trunc(even/2)

3-1   q:=even-p

3-2   判断p是否为素数（pyn:=1表示p为素数，pyn:=0表示p不是素数）

3-3   判断q是否为素数（qyn:=1表示q为素数，qyn:=0表示q不是素数）

3-4   IF pyn=1 AND qyn=1

    THEN writeln(even,’=’,p,’+’,q)

二级求精：

1、read(even)

2、IF even MOD 2=0

    THEN 第3步

3-2、n:=2

WHILE n<sqrt(p) DO

         BEGIN

           n:=n+1;

           IF p MOD n=0

             THEN pyn:=0;

         END;

3-3、n:=2

WHILE n<sqrt(q) DO

         BEGIN

           n:=n+1;

           IF q MOD n=0

             THEN qyn:=0;

         END;

PROGRAM T75\_008;

  VAR

   even,p,q,n,pyn,qyn:integer;

  BEGIN

   write('Please input a even:');

   read(even);

    IF even MOD 2=0

     THEN BEGIN

            FOR p:=2 TO trunc(even/2) DO

              BEGIN

                q:=even-p;

                n:=1;

                pyn:=1;

                qyn:=1;

                WHILE n<sqrt(p) DO

                  BEGIN

                    n:=n+1;

                    IF p MOD n=0

                      THEN pyn:=0;

                  END;{WHILE-1}

                n:=1;

                WHILE n<sqrt(q) DO

                  BEGIN

                    n:=n+1;

                    IF q MOD n=0

                      THEN qyn:=0;

                  END;{WHILE-2}

                IF (pyn=1) AND (qyn=1)

                  THEN writeln(even,'=',p,'+',q);

              END;{FOR}

          END{IF}

     ELSE writeln('You must input a even!');

  END.

T75\_009  打印杨辉三角形

1

1   1

1  2   1

1  3  3   1

1  4  6  4   1

……           ……

1  10  45 ……   45 10  1

讨论：杨辉三角形中，每个数据值可以由组合 来计算。上列杨辉三角形可以写成：

……                       ……

     ……     ……

已知

可以由递推公式计算

为了输出三角形，我们将中心线位置定在40格处。每一行的第一个值应往左移3格，最大数是3位整数，每个数之间留3个空格，因此输出值的场宽可以定义为6。

这是一个二重循环问题，且内循环n的次数依赖于外循环m（对应每行输出不同的个数）。

一级算法

FOR m:=0 TO 10 DO

  BEGIN

   1、输出每行的第一个值，左面留空

   2、输出每行的其余值

    3、换行

  END

二级求精

第1步  输出每行的第一个值，左面留空

c:=1

write(c:40-3\*m)

第2步  输出每行的其余值

FOR n:=1 TO m DO

  BEGIN

   c:=c\*(m-n+1) DIV n

   write(c:6)

  END

第9课PASCAL编程基本知识复习20080407

一、填空题：

1、标识符是\_\_**以字母开头的字母、数字组合**\_\_。标识符可用来表示\_**常量**\_、\_\_**变量**\_\_、**类型**、**文件**、**函数**、**过程**、**程序**的名字。标识符的长度是没有限制的。但标准PASCAL规定能区分的有效标识符长度是\_\_**8**\_\_。标识符的选取最好\_\_**有一定的含义**\_\_，这样便于记忆，也增加了程序的可读性。（5）

2、尽可能多地写出你知道的函数名称及功能（20）

|  |  |  |  |
| --- | --- | --- | --- |
| 函数 | 功能 | 函数 | 功能 |
| 例：sqrt() | 开方 | **round( )** | **四舍五入，取整数** |
| **abs( )** | **绝对值** | **exp( )** | **以e为底的指数** |
| **sqr( )** | **平方** | **ln( )** | **以e为底的自然对数** |
| **ord( )** | **取序号** | **sin( )** | **正弦函数** |
| **chr( )** | **取字符** | **cos( )** | **余弦函数** |
| **pred( )** | **前导字符** | **arctan( )** | **余切函数** |
| **succ( )** | **后继字符** |   |   |
| **odd( )** | **奇函数** |   |   |
| **trunc( )** | **去掉小数部分** |   |   |

3、写出四种标准数据类型标识符（4）

**实型   real            整型   integer**

**字符型 char            布尔型  boolean**

4、从程序的结构看，PASCAL程序的四种基本结构包括：\_**顺序结构**\_、\_\_**选择结构**\_、\_\_**循环结构**\_\_、函数与过程结构。（3）

5、Pascal程序可以分为三个部分，请将这三个部分的名称填入下面括号中（3）

|  |
| --- |
| PROGRAM circle(input,output);  CONSTPI=3.14159;  VARr,l,s:real;  BEGINread(r);l:=2\*PI\*r;s:=PI\*r\*r;write(r,l,s);  END. |

|  |
| --- |
| （ **程序首部** ） |

|  |
| --- |
| （ **说明部分**  ） |

|  |
| --- |
| （ **语句部分** ） |

6、在Pascal中，（15）

有六个算术运算符，它们是\_\_**+、-、\*、/、MOD、DIV**\_\_\_。

有六个关系运算符，它们是\_\_**>、>=、<、<=、=、<>**\_\_。

有三个逻辑运算符，它们是\_\_**AND、OR、NOT**\_\_\_。

|  |
| --- |
| **优先级由高到低** |

|  |
| --- |
| **（）** |

|  |
| --- |
| **函数** |

|  |
| --- |
| **NOT** |

|  |
| --- |
| **AND****\*****/****div****mod** |

|  |
| --- |
| **OR****+****-** |

|  |
| --- |
| **<****<=****=****>=****>****<>**  |

7、请自己设计一张图表将第6题中的运算符，再加上函数和括号在PASCAL程序中的运算优先规则表示出来。（3）

8、请将下列实型数据用另一种表示法表示（6）

|  |  |  |  |
| --- | --- | --- | --- |
| 小数表示法 | 科学表示法 | 小数表示法 | 科学表示法 |
| 12.5 | **1.25e+1** | **100** | 1e+2 |
| 132.67 | **1.3267e+2** | **-30.5** | -3.05e+1 |
| 0.0025 | **2.5e-3** | **-0.0002509** | -2.509e-4 |

9、将下列代数式写成Pascal的表达式（4）

（1）a-(m-d)×4÷b+3y=A2                 （2）

|  |
| --- |
| **a-(m-d)\*4/b+3\*y=A2** |

|  |
| --- |
| **(p\*q\*sqr(r+1))/(sqr(r+1)-1)** |

|  |
| --- |
| **abs(3-exp(x)\*ln(1+x))** |

|  |
| --- |
| **(88-16)/(47-15)-4\*a** |

（3）                         （4）|3-exln(1+x)|

二、程序找错（每个程序有4处错误，请你找出来）

10、

PROGRAM T55\_002;

|  |
| --- |
| **错误一：第2行少“VAR”；****错误二：第5行少分号；****错误三：第4、6、8行，字符串应用单引号，题中用了双引号；****错误四：第8行字符串与变量之间少逗号。** |

    a,h:integer;

  BEGIN

   write(“San Jiao Xing Di:”);

   readln(a)

   write(“San Jiao Xing Gao:”);

   readln(h);

   writeln(“San Jiao Xing Mian Ji:”a\*h/2:6:1)

  END.

11、

PROGRAM T65\_004;

  VAR

   n,integer;

  BEGIN

   write('Please input a number from 0 to 6 for weekday:')

   read(n);

    CASE n

     0:writeln('Sunday');

|  |
| --- |
| **错误一：第2行变量与数据类型之间应用冒号，这里用了逗号；****错误二：第5行少分号；****错误三：第7行，CASE语句少了OF；****错误四：CASE语句没有END与之配对。** |

     1:writeln('Monday');

     2:writeln('Tuesday');

     3:writeln('Wednesday');

     4:writeln('Thursday');

     5:writeln('Friday');

     6:writeln('Saturday');

  END.

12、PROGRAM T75\_002;

  VAR

   ch:char;

   ZiMu,ShuZi,FuHao:integer;

  BEGIN

   ZiMu:=0;

   ShuZi:=0;

   FuHao:=0;

    WHILE ch<>'?'

     BEGIN

       read(ch);

|  |
| --- |
| **错误一：第9行WHILE语句少DO；****错误二：第12行IF语句中，字符没有加单引号；****错误三：第13行，赋值语句不对，应为:=****错误四：第14行，IF语句中，AND两边的逻辑判断应各自加括号，即（ord(ch)>=48） AND （ord(ch)<=57）。** |

       IF (ch>A) AND (ch<Z) OR (ch>a) AND (ch<z)

         THEN ZiMu=ZiMu+1

         ELSE IF ord(ch)>=48 AND ord(ch)<=57

           THEN ShuZi:=ShuZi+1

           ELSE FuHao:=FuHao+1;

     END;

     writeln('Zi Mu Ge Shu: ',ZiMu);

     writeln('Shu Zi Ge Shu: ',ShuZi);

     writeln('Qi Ta Fu Hao Ge Shu: ',FuHao)

    END.