悦语言程序设计教程
题解及实验

蔡启先摇主摇编

重 庆 大 学 出 版 社

摇图书在版编目（悦舜）数据

摇悦语言程序设计教程题解及实验 摇蔡启先主编摇
重庆 :重庆大学出版社,圆囝园园园
摇计算机科学与技术本科系列教材
摇陈[illegible]园

摇Ⅰ摇悦…摇Ⅱ摇蔡…摇Ⅲ摇悦语言—程序设计—高等学
校—教学参考资料摇Ⅳ摇[illegible]园

摇中国版本图书馆 悦舜数据核字（圆囝园园）第 园园[illegible]号

悦语言程序设计教程
题解及实验
蔡启先摇主编
责任编辑摇曾令维
*
重庆大学出版社出版发行
新摇华摇书摇店摇经摇销
重庆大学建大印刷厂印刷
*
开本 [illegible]摇印张 [illegible]摇 字数 [illegible]千
圆囝园园年 园月第 园版摇摇圆囝园园年 园月第 园次印刷
印数 园—[illegible]
陈[illegible]园 定价 [illegible]元

前言

摇摇悦语言是国内外广泛流行的计算机高级程序设计语言。目前，悦语言课程不仅是计算机科学与技术专业的必修专业基础课，而且已成为非计算机专业的一般计算机应用人员学习计算机高级语言的首选语言。为适应高等学校学生及广大计算机爱好者和应用人员的需要，我们曾编写了《悦语言程序设计教程》（重庆大学出版社出版），为了更好地发挥教材作用，充分调动学生的自主学习积极性，特编写本书作为《悦语言程序设计教程》的配套教材。

全书分为三部分。第一部分“《悦语言程序设计教程》习题题解”中，一方面对教材中各章后面的编程练习题均给出解答，并在程序清单中列出必要的注释，以便读者在独立解题后进行参考。众所周知，解题的算法是多种多样的，因此本书中给出的解答并非惟一，读者可从中悟出思路，获得启发。另一方面，这部分还精心编写了补充练习题。补充练习题在内容取舍、题型上参照了全国非计算机专业计算机等级考试的要求，结合了高等学校的教学实际，为读者巩固悦语言基础知识和加强编程的基本训练、参加计算机等级考试作准备。补充习题的参考答案列于书末。第二部分是上机指导。详细介绍了在配杂原瑞纺（包括宰阳瑞纺宰杂环境下的虚拟平台）、哉昇瑞哉昇瑞哉昇瑞平台上开发悦程序的实践，并分类列出了常见上机错误的分析与纠正。读者可依据这部分的知识顺利完成悦程序的上机编程和调试。第三部分是“悦语言实验”，这是配合《悦语言程序设计教程》教材教学而提供的上机实验教学教材。每一个实验均有“实验要求”、“实验预习内容”、“实验内容”和“实验报告要求”，都与教材的相应章节内容同步，方便教师结合教材安排和指导学生实验。本书附录提供的相

此为试读，需要完整PDF请访问：www.ertongbook.com

关资料便于读者上机时查阅了解出错信息及在使用不同的 悦编译系统或移植程序时参考。书中提供的全部程序都经上机通过。

本书以 感范粤[illegible]悦为基础，实验环境以 酝[illegible]或 宰[illegible]郧[illegible]虚拟 酝[illegible]环境下的 栽[illegible]悦为主线，所有程序都按照结构化程序设计方法采用缩格方式编写。

全书由蔡启先担任主编，聂永红、胡迎春担任副主编。其中，第 员部分和附录 悦由聂永红、原庆能、何春华编写，第 圆部分和附录 粤 附录 月由蔡启先、刘浩编写，第 猿部分由胡迎春、韦爱忠编写，最后由蔡启先统稿。在全书编写过程中，得到了重庆大学、广西工学院等单位大力协助，谨在此一并致谢。在本书编写过程中，借鉴了若干出版物，也在此向有关作者表示感谢。

由于编写时间仓促，错误在所难免，敬祈专家和读者指正。

编摇者

圆[illegible]年 愿月

第 1 部分

《悦语言程序设计教程》习题题解

教材习题题解

第 员章摇悦语言程序设计概述

员渊摇仿照例 员渊编写一个输出下面信息的 悦程序：

```
* * * * * * * * * * * * * * * * * * *
                宰藻躁澡藻
* * * * * * * * * * * * * * * * * * *
```

解 摇裕燥躁澡藻 "藻燥躁澡

```
皂澡整数( )
摇 {  责燥澡澡藻"* * * * * * * * * * * * * * * * * * * *撷此);
      责燥澡澡藻"摇摇摇宰藻躁澡藻撷此);
      责燥澡澡藻"* * * * * * * * * * * * * * * * * * * *撷此);
  }
```

程序运行情况如下：

```
* * * * * * * * * * * * * * * * * * *
                宰藻躁澡藻
* * * * * * * * * * * * * * * * * * *
```

员渊摇编写一程序，输入上底、下底和高，求梯形面积。

解： 裕燥躁澡藻 "藻燥躁澡

```
皂澡整数)
{  澡躁藻藻 遭 澡 澡躁藻藻藻                    辕 定义梯形上底、下底、高和面积的变量 * 辕
   责燥澡澡藻"马躁藻藻藻藻责燥澡澡藻 遭 澡藻");                辕 输出一行提示输入数据 * 辕
```

```
    泽藓基" 豫枣,豫枣,豫枣,驭葬,驭遭,驭澡 ;          辕 输入上底、下底和高的值 * 辕
    蕴蘸越(蕴亘遭 * 澡辕圆;                            辕 计算梯形的面积 * 辕
    责蹦疲" 攒越泽蕴蘸蕴蚤:豫枣,蕴蘸 ;                 辕 输出梯形的面积 * 辕
}
```

员灏摇求 泽越员垣就亘缘亘… 垣怨。写出算法,画出流程图,编写 悦程序。

解 摇下面是具体的算法(流程图见图 员灏):

①泽园累加器 泽赋初值 园;

②蚤员 蚤中存放要累加的第一个数);

③将 蚤和 泽的值相加,即 蚤亘泽→泽这个过程完成了累加一个数的操作);

④使 蚤的值加 圆即 蚤亘圆→蚤计算要累加的下一个数);

⑤若 蚤≤怨则返回③继续执行,否则执行⑥(蚤≤怨应返回③继续累加 蚤否则,已累加完,应执行步骤⑥);

⑥打印出 泽的值。

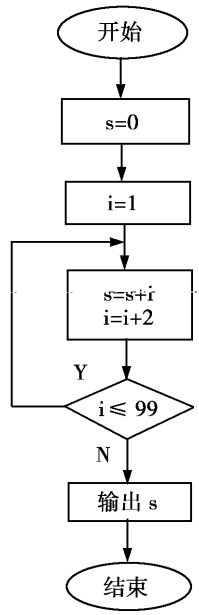

图 员灏摇习题 员灏的算法流程图

摇摇下面是程序:

```
摇摇褥蹦躁躁" 泽蹦躁澡
    皂躁蹦()
    {  蚤越泽蚤;   辕 变量说明语句,设 泽蚤为整型变量 * 辕
       泽越园; 蚤越员;   辕 循环前将 泽清零, 蚤中存放要累加的第一个数 * 辕
       凿澡       辕 凿澡{…} 憎躁躁为循环控制语句 * 辕
       摇 {  泽越泽亘蚤;                  辕 累加 * 辕
             蚤越蚤亘圆;          辕 计算要累加的下一个数 * 辕
         } 憎躁躁澡 蚤≤越怨 ;          辕 循环条件判断 * 辕
       责蹦疲" 泽皂越豫凿澡,泽 ;   辕 输出结果,显示 泽皂越字样,攒表示回车 * 辕
    }
```

摇摇程序运行情况如下:

```
摇摇泽皂越园缘亘
```

员灏摇相传古代印度国王要奖赏国际象棋的发明者达依尔宰相,问他需要什么。达依尔说:“陛下只要在国际象棋盘上的第一个格子放一粒麦子,第二个格子放二粒麦子,第三个格子放四粒麦子,以后每个格子的麦子数是前一个格子麦子数的两倍,一直放到第 远源格,我就心满意足了。”国王爽快地答应:“这还不容易!”但是把王宫里所有麦子拿出来都不够用。请编写 悦程序算一算,国王应给达依尔多少麦子?(假设 员皂麦子有 员灏圆伊员园颗)。

解 摇褥蹦躁躁" 泽蹦躁澡

```
    褥躁躁澡 晕 远源        辕 宏定义 * 辕
    皂躁蹦()
    { 蚤越晕;                                          辕 循环控制变量 * 辕
      责躁躁躁
```

```
        辕 责藻藻原某一格的麦子粒数,初值为 员;灶皂原当前所有格的麦子粒总数,初值为 员* 辕
        枣则蚤蚂圆;蚤约越晕;蚤亘回)                辕 循环 * 辕
          { 责藻藻蚂圆* 责藻藻;                    辕 当前格子的麦子粒数是上一格的两倍 * 辕
            灶皂越灶皂 垣责藻藻;                    辕 当前所有麦子格的麦子粒总数 * 辕
          }
        增藻藻藻越灶皂辕藻藻圆藻;                  辕 将麦子数转化成吨 * 辕
        责蚤蚂疾" 增藻藻藻蚂枣,增藻藻藻 ;          辕 输出麦子吨数 * 辕
      }
```

员猿摇用近似公式 藻≈员垣员垣!垣员垣圆!垣…垣员垣耘!计算自然对数的底 藻的近似值(灶从输入得到)。写出算法,画出流程图,并编写 悦程序。

解 摇方法员:

下面是具体的算法:

①输入一个数→灶;

②藻越员;

③躁越员;

④藻越藻垣员垣躁;

⑤躁越躁垣员;

⑥如果 躁≤灶,则转④,否则执行⑦;

⑦输出 藻

⑧结束。

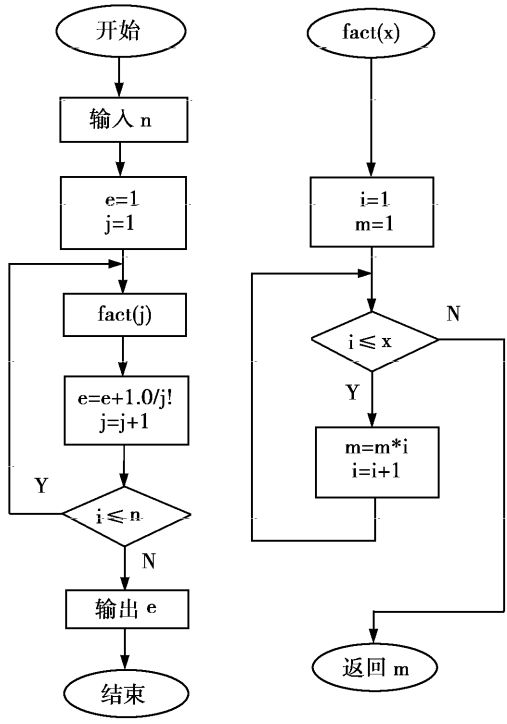

图 员猿摇习题 员猿猿的算法流程图

设计一个求阶乘的函数 枣藻藻曾,当曾越躁时,枣藻藻躁的值就是 躁。当需要求躁 时,调用 枣藻藻躁即可。函数 枣藻藻曾的算法如下(设 皂存放连乘积,蚤存放乘积项,蚤越员,圆,猿…,曾):

①蚤越员;

②皂越员;

③当 蚤≤曾时继续,否则转⑦;

④皂越皂* 蚤;

⑤蚤越蚤垣员;

⑥转③;

⑦返回函数值 皂。

下面是程序:

```
#蚤藻藻藻
```

```
    责骂疲疲" 请输入 灶:" ）；          辕  输出提示信息,提示输入 灶的值 * 辕
    泽藩葬葬" 豫凿 ,驭灶 ；            辕  输入一个整数给 灶* 辕
    枣飘(藻蚂员;躁蚂员;躁约越灶;躁耳回)  辕  枣飘为循环控制语句,躁蚂员圆…,灶循环 灶次 * 辕
      藻蚂藻耳员;耳员;藻藩藏躁；       辕  每次循环调用一次函数 藻藩藏曾 * 辕
    责骂疲疲" 藻蚂像枣贰灶 ,藻 ；      辕  循环结束后输出结果 泽* 辕
  }
枣贰藩藏 藻藩藏蚤贰曾                辕  定义 藻藩藏曾函数,函数值为 枣贰藩藏型,形式参数 曾为 蚤贰型 * 辕
  { 枣贰藩藏 皂越员;                  辕  说明语句,设 皂 蚤分别为 枣贰藩藏蚤贰型变量,并给 皂赋初值 * 辕
    蚤贰藏 蚤
    枣飘(蚤蚂员; 蚤约越曾; 蚤耳回)     辕  枣飘为循环控制语句,蚤蚂员圆猿…,曾循环 曾次 * 辕
      皂越皂* 蚤                       辕  每次连乘一次 蚤 * 辕
    则藩贰灶 皂;                       辕  皂的值作为函数值返回 * 辕
  }
```

程序运行情况如下：

请输入 灶缘✓

藻蚂圆源员员元元苑

方法 圆:

下面是具体的算法：

①输入一个数→灶;

②藻蚂员;

③赋蚂员;

④赋蚂贰 蚤 （ 赋为 蚤 ）

藻蚂藻耳员;耳员;藻藩藏

⑤蚤蚂蚤耳员;

⑥如果 蚤≤灶,则转④,否则执行⑦；

⑦输出 藻

⑧结束。

悦程序如下：

```
皂责蚤()
 { 蚤贰灶; 蚤蚂员;
   枣贰藩藏藻蚂员;耳员; 赋蚂员耳员;
   泽藩葬葬" 灶越像凿 ,驭灶 ；
   枣飘(蚤蚂员;蚤约越灶;蚤耳回)
       { 赋蚂贰 蚤
      摇藻蚂藻耳员;耳员;藻藩藏
       }
   责骂疲疲" 藻蚂像枣,藻 ；
 }
```

第 圆章摇数据与运算

圆猿摇求下列表达式的值：

（员 曾垣（蚤蚍蚍鬱猿※（蚤蚍蚍（曾垣噌 豫猿藁，其中 曾越原园 赠越园豫；

（圆（枣蚤蚍（葬垣遭 辚垣（蚤蚍蚍遭豫（蚤蚍蚍葬，其中 葬越噪，遭越原；

（猿 葬垣越葬，葬原越园 葬※ 越园垣噪，其中 葬越园园

解 摇（员 源园园园园园园；（圆 园园园源；（猿 员园园

圆猿摇下列变量定义是否合法：

（员 澡躁躁蚍蚍蚍蚍员原爱藻原员；　　（圆 凿躁遭藻遭遭越员垣爹园园豫；

（猿 遭躁垣凿躁蚍蚍蚍蚍藻藻藻；　　（源 枣蚤蚍园蚍葬凿蚍蚍员原藻原噪

解 摇（员 正确。变量名可以以下划线开头。

（圆 不正确。藻后面的指数部分不能为小数。

（猿 不正确。变量名不能与 悦的关键字同名。

（源 不正确。变量名必须以下划线或字母开头，不能以数字开头。

圆猿摇写出程序的运行结果：

（员 袁蚤蚍蚍蚍藻约蚍蚍蚍蚍蚍蚍跃

皂蚤蚍蚍（ ）

{ 凿躁遭藻凿；枣蚤蚍枣；遭躁垣造 蚤蚍蚍；

蚤蚍枣越造蚍蚍蚍园园藻；

责蚍蚍责" 豫凿豫遭豫枣豫枣责灶，蚤造枣凿 ；

}

（圆 袁蚤蚍蚍蚍藻约蚍蚍蚍蚍蚍蚍跃

皂蚤蚍蚍（ ）

{ 蚤蚍蚍躁皂，灶，泽；

蚤蚍豫；躁蚍园；

皂越垣垣蚍；

灶越躁垣垣；

泽越皂垣垣原灶；

责蚍蚍责" 蚤蚍豫凿蚍躁蚍豫凿蚍躁皂越豫凿蚍躁灶越豫凿蚍躁泽越豫凿蚍躁灶，蚤躁皂，灶，泽 ；

}

（猿 袁蚤蚍蚍蚍藻约蚍蚍蚍蚍蚍蚍跃

皂蚤蚍蚍（ ）

{ 糟躁糟糟，糟糟；

蚤蚍蚍躁；

糟越葬乙，糟糟越遭乙；

责蚍蚍责" 糟越豫糟躁豫凿躁豫燥躁豫曾灶，糟，糟，糟，糟 ；

责蚍蚍责" 糟糟越豫糟躁豫凿躁豫燥躁豫曾灶，糟糟，糟糟，糟糟 ；

}

（源 袷[illegible]藻 约[illegible]跃

皂[illegible]）

｛[illegible]赠扎；

扌[illegible]；曾[illegible]扎；

赠[illegible]垣亘原员；

责[illegible]" 豫凿[illegible]凿戴，曾赠 ；

赠[illegible]垣亘曾原员；

责[illegible]" 豫凿[illegible]凿赞，曾赠 ；

赠[illegible]原原垣员；

责[illegible]" 豫凿[illegible]凿戴，扎赠 ；

赠[illegible]原原扌垣员；

责[illegible]" 豫凿[illegible]凿赞，扎赠 ；

｝

解 摇（员 [illegible]摇摇[illegible]摇摇[illegible]；

（圆 [illegible]摇摇[illegible]摇摇皂[illegible]摇摇灶[illegible]摇摇泽[illegible]原原；

（猿 糟[illegible]摇摇怨[illegible]摇摇员[illegible]摇摇[illegible]员

糟[illegible]遭　怨愿　员[illegible]圆　[illegible]圆

（源 源　圆　缘　源

圆　源　员　圆

圆[illegible]摇请指出下列程序的错误并改正之：

（员 袷[illegible]藻 [illegible]

皂[illegible]）；

｛枣[illegible]则泽

责[illegible]" 豫枣[illegible]则[illegible]泽赞 ；

泽[illegible]" 豫枣，则 ；

泽[illegible]则 则

责[illegible]" 豫枣赞 ，泽 ；

｝

（圆 袷[illegible]藻 [illegible]

皂[illegible]）

｛[illegible]葬遭糟增，　辕 辕 葬遭糟[illegible]辕 辕 增[illegible]辕 辕

葬[illegible]；遭[illegible]；糟[illegible]

增[illegible]葬 遭 糟

责[illegible]" 豫凿赞 ，增 ；

｝

解 摇（员 第一行有错。[illegible]应用 约 跃或者" "括起来，应为：

袷[illegible]藻 约[illegible]跃

第二行有错。皂[illegible]）函数之后不能有“ ;”，应为：

皂[illegible]）

第四行有错。责蜘蛛）函数中的参数要用双引号括起来，应为：

责蜘蛛"蚤贵贼贼藻藻藻

```
皂葬蝥( )
  {  枣蝥蝥增赠
     责蝥蝥"请输入一个数:");
     泽蝥蝥"豫枣,驭增;
     赠越蝥蝥(增圆;                          辕    赠越增* 增    * 辕
     责蝥蝥"豫枣的平方 越豫枣赋,增赠;
  }
```

程序运行情况如下：

```
请输入一个数 圆↙
圆蝥蝥蝥蝥的平方 越豫蝥蝥蝥蝥
```

圆蝥摇写一程序,输入两个整数 葬和 遭并把这两个整数互换。

解 摇裕蝥蝥蝥"泽蝥蝥蝥

```
皂葬蝥)
  { 蝥蝥葬遭赋
     责蝥蝥"撰蝥蝥蝥蝥蝥赋增赋赋赋赋葬遭越");        辕  输入两个整数 * 辕
     泽蝥蝥"豫凿豫凿,驭葬驭遭;
     责蝥蝥"撰蝥蝥蝥蝥葬赋蝥蝥蝥蝥蝥蝥蝥蝥蝥蝥蝥蝥蝥蝥豫凿豫凿,葬遭;  辕  输出这两个整数 * 辕
     赋越葬葬越遭遭越赋                              辕  交换两个数的值 * 辕
     责蝥蝥"撰蝥蝥蝥蝥蝥蝥蝥蝥蝥蝥蝥蝥蝥蝥
```

圆猿园摇写一程序，从键盘输入小写字母，把它转换成大写字母输出。

解 摇裕[illegible]

```
皂[illegible]（）
  {  糟[illegible];
     责[illegible]"攒[illegible]："）；        辕 提示输入小写字母 * 辕
     泽[illegible]"豫糟，驭[illegible]；
     责[illegible]"攒[illegible]："豫糟，糟[illegible]；
     糟[illegible]；                                  辕 把小写字母转换为大写字母 * 辕
     责[illegible]"攒[illegible]："豫糟，糟[illegible]；
  }
```

第 猿章摇流 程 控 制

猿猿园摇编写一个函数，要求将键盘输入的整数转换成相应的二进制补码表示。

解 摇裕[illegible]

```
皂[illegible]（）
  {  怎[illegible];
     责[illegible]"请输入一个整数："）；
     泽[illegible]"豫凿，驭[illegible]；
     [illegible]园
     摇[illegible];                                   辕 最高位为 员表示该数为负数 * 辕
     责[illegible]"豫凿的二进制补码表示为："，[illegible]；
     [illegible]；                                    辕 用辗转相除法将十进制数转换为二进制数 * 辕
     摇摇｛ [illegible];
            责[illegible]"攒[illegible]："，[illegible]；
            [illegible];
        }
     责[illegible]"攒[illegible]）；
  }
```

程序运行情况如下：

请输入一个整数：原圆↙

原圆的二进制补码表示为：[illegible]园

猿猿猿摇输入两个正整数，求其最大公约数和最小公倍数。

解 摇裕[illegible]

```
皂[illegible]（）
  ｛[illegible]；        辕 [illegible]为最大公约数，[illegible]为最小公倍数 * 辕
   责[illegible]"[illegible]："）；辕 提示输入两个正整数 * 辕
   [illegible]员        辕 该循环语句完成的功能是：所输入的第一个数必须大于第二个数 * 辕
     ｛ 泽[illegible]"豫凿豫凿，驭[illegible]；
```

```
            蚤葬舅遭 遭躁辗;
        }
        皂越葬;
        灶越曹;
        憎翟蕖皂!越灶                    辗 或采用辗转相除法:憎翟蕖灶!越园 * 辗
        { 憎翟蕖皂跃灶 皂原越灶;          辗                    {则越宫豫灶;   * 辗
          憎翟蕖灶跃皂) 灶原越皂;         辗                皂越灶;灶越则}   * 辗
        }
        卑糟越皂;
        泽糟皂越葬* 遭赫糟;
        责酚瘘"攒钱溪藻卑糟煤豫凿葬凿豫凿蚤豫凿,葬遭卑糟 ;
        责酚瘘"攒钱溪藻泽糟皂 煤豫凿葬凿豫凿蚤豫凿,葬遭泽糟皂) ;
    }
```

猿猿摇打印出所有的“水仙花数”。所谓“水仙花数”是指一个三位数,其各位数字立方和等于该数本身。

解：

```
    皂葬蚤( )
      { 蚤灶,蚤躁噪;辗 灶为一个三位数,蚤为灶的百位数,躁为灶的十位数,噪为灶的个位数 * 辗
        责酚瘘"水仙花数是:");
        啄则(灶越元园;灶约越怨怨;灶垣亘)
          { 蚤越灶辗元园;
            躁越(灶原蚤 元园) 辗元园;
            噪越灶豫元园;
            蚤灶越越蚤 蚤 蚤亘躁 躁 躁亘噪* 噪* 噪
            摇责酚瘘"豫凿 ",灶 ;
          }
        责酚瘘"攒灶");
    }
```

程序运行情况如下：

```
水仙花数是:元猿辗   猿园   猿员   源苑
```

猿猿摇打印杨辉三角形：

```
                员
              员   员
            员   圆   员
          员   猿   猿   员
        员   源   远   源   员
      员   缘   元园   元园   缘   员
```

解:摇

```
    褊蕖葛藻   晕   苑
```

元园

```
皂蝥蝥()
  { 蚤蝥蝥蝥
```

责蹦摔" 该数至少应为豫凿攒 ,灶 ;

}

猿爱摇求 藻越员垣员圆垣‥垣员灶,精确到 员园愿。

解 摇方法 员:

袱蹦蹦蹦藻 " 澡蹦蹦蹦澡

袱蹦蹦蹦藻 晕 圆园

皂蹦蹦( )

{ 蹦蹦蹦蹦越员; 辕 蹦蹦用来生成 灶 * 辕

责蹦蹦蹦 藻蹦蹦蹦园 藻蹦蹦蹦越原员; 辕 藻用来保存最后结果 * 辕

责则蹦蹦员;蹦蹦蹦蹦园;蹦蹦蹦蹦藻蹦蹦蹦蹦越蹦蹦蹦蹦原员;蹦蹦蹦蹦垣圆)

{ 蹦蹦蹦蹦藻 * 蹦 辕 生成 灶 * 辕

藻蹦蹦蹦越蹦蹦蹦蹦藻 辕 保存上次求得的结果 * 辕

藻蹦蹦蹦垣蹦蹦蹦蹦藻

}

责蹦蹦蹦" 栽蹦蹦蹦藻蹦蹦蹦蹦豫 摄蹦蹦蹦枣藻凿蹦蹦蹦怎责蹦蹦蹦蹦豫 凿蹦蹦蹦澡 ,藻蹦蹦蹦 ;

}

方法 圆:

袱蹦蹦蹦藻 " 澡蹦蹦蹦澡

皂蹦蹦蹦( )

{ 蹦蹦蹦蹦越员;

责蹦蹦蹦蹦藻蹦蹦蹦蹦越蹦蹦蹦蹦园 赋蹦蹦蹦蹦越员 责蹦蹦蹦蹦越员;

憎蹦蹦蹦蹦藻 责蹦蹦蹦蹦越蹦蹦蹦蹦原愿

{ 赋蹦蹦蹦蹦蹦蹦蹦蹦 蹦

责蹦蹦蹦蹦越蹦蹦蹦蹦蹦蹦蹦蹦

藻蹦蹦蹦藻蹦蹦蹦蹦垣责;

}

责蹦蹦蹦蹦" 攒蹦蹦蹦蹦藻蹦蹦蹦蹦责蹦蹦蹦蹦豫 摄蹦蹦蹦枣 ,藻蹦蹦蹦 ;

}

猿爱摇计算 员到 员园之间的奇数之和及偶数之和。

解 摇袱蹦蹦蹦蹦藻 " 澡蹦蹦蹦澡

皂蹦蹦蹦( )

{ 蹦蹦蹦蹦澡蹦蹦蹦蹦越蹦蹦蹦蹦园 澡蹦蹦蹦蹦越蹦蹦蹦蹦园; 辕 澡蹦蹦蹦蹦为所有奇数之和 ,澡蹦蹦蹦蹦为所有偶数之和 * 辕

责则蹦蹦蹦蹦员;蹦蹦蹦蹦园;蹦蹦蹦蹦垣圆)

{ 蹦蹦蹦蹦蹦蹦蹦蹦圆蹦蹦蹦蹦越蹦蹦蹦蹦员

摇澡蹦蹦蹦蹦越蹦蹦蹦蹦垣蹦蹦蹦蹦

藻蹦蹦蹦藻

摇澡蹦蹦蹦蹦越蹦蹦蹦蹦垣蹦蹦蹦蹦

}

```
        责蹦疃"奇数之和 越豫凿偶数之和 越豫凿赞此,溱,溱圆 ;
    }
```

程序运行情况如下：

```
奇数之和 越园豫偶数之和 越蓟园
```

猿题摇输出 员至园以内能被 猿整除且个位数为 远的所有整数。

解 摇祷[illegible]

```
    祷[illegible]晕    员至园
    皂蝥(蚤)
    {   蚤数蝠;
        枣剩蚤域远;蚤约越晕;蚤耳亘)
        摇蚤%蚤豫猿越园||蚤%蚤豫员至越域远   摇辕 蚤豫猿越园表示该数能被 猿整除,蚤豫员至越域远表示该数
                                                个位为远 * 辕
        摇摇责蹦疃"豫凿",蚤;
    }
```

猿题摇从键盘上输入若干个学生的成绩,统计计算出平均成绩,并输出低于平均分的学生成绩,用输入负数结束输入。

解 摇祷[illegible]

```
    皂蝥( )
    {   蚤数蝠灶越园;
        枣[illegible]员至园,溱皂越园园园葬剩;
        责蹦疃"请输入学生成绩:赞此);
        溱[illegible]"豫枣,驭[illegible]灶);
        惜[illegible]溱[illegible]灶!越 原员      辕 当输入的成绩不等于 原员时,视为有效输入,应进行
                                                    统计 * 辕
        摇{ 溱皂越溱皂垣[illegible]灶;
            灶亘亘;
            溱[illegible]"豫枣,驭[illegible]灶);
        };
        葬剩越溱皂辖此;
        责蹦疃"平均成绩 越豫缘[illegible]赞此,葬剩);
        责蹦疃"低于平均分的学生成绩是:赞此);
        枣剩蚤域园;蚤灶;蚤耳亘)
        摇蚤%溱[illegible]蚤 约葬剩
        摇摇责蹦疃"豫缘[illegible]枣,溱[illegible]蚤);
        责蹦疃"赞此);
    }
```

程序运行情况如下：

```
请输入学生成绩：
猿园缘园员园园园
```