

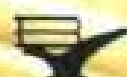
21世纪计算机课程系列教材



# FORTRAN 90

## 程序设计上机指导 与习题选解

主编 / 刘卫国 副主编 / 蔡旭晖



北京邮电大学出版社

# 21世纪计算机课程系列教材



责任编辑 / 陈露晓  
封面设计 / 殷健工作室

ISBN 7-5635-0661-6



9 787563 506613 >



ISBN 7-5635-0661-6/TP·90

定价：18.00 元

TP312  
1070A

21 世纪计算机课程系列教材



# FORTRAN 90

## 程序设计上机指导 与习题选解

主编 / 刘卫国 副主编 / 蔡旭晖

北方工业大学图书馆



00529340



北京邮电大学出版社

## 内 容 简 介

学习程序设计,上机实践是十分重要的环节。本书是与《FORTRAN 90 程序设计教程》配套的实验教材,主要有五方面的内容:Fortran PowerStation 4.0 集成开发环境、上机实验指导、常用算法设计、程序调试方法和习题选解。前两部分内容帮助读者熟悉上机环境,方便读者上机练习。通过有针对性的上机实验,可以更好地熟悉 FORTRAN 90 程序设计的方法;第三、四部分内容通过总结编程思路,介绍程序调试的一般方法,来提高读者的程序设计能力和调试程序的能力;第五部分内容可以作为课程学习或参加各种计算机考试的辅导材料。

### 图书在版编目(CIP)数据

FORTRAN 90 程序设计上机指导与习题选解/刘卫国主编. —北京:北京邮电大学出版社,2003

ISBN 7-5635-0661-6

I. F... II. 刘... III. FORTRAN 语言—程序设计—高等学校—教学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 006114 号

---

书 名: FORTRAN 90 程序设计上机指导与习题选解

主 编: 刘卫国

责任编辑: 陈露晓

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号(100876)

电话传真: 010-62282185(发行部)010-62283578(FAX)

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷: 国防科技大学印刷厂印刷

开 本: 787mm×1 092mm 1/16

印 张: 12.25

字 数: 280 千字

版 次: 2003 年 2 月第 1 版 2003 年 2 月第 1 次印刷

---

ISBN 7-5635-0661-6/TP·90

定 价: 18.00 元

如有质量问题请与发行部联系

版权所有 侵权必究

# 前 言

学习高级语言程序设计应掌握高级语言的语法规则,掌握程序设计的基本方法和技巧。掌握语法规则要严格、准确,但不应死记硬背,可以通过对比分析、上机练习来加强理解和记忆,而程序设计能力更要通过大量的上机实践来培养。本书是与《FORTRAN 90 程序设计教程》配套的实验教材。主要介绍 5 方面的内容:

1. Fortran PowerStation 4.0 集成开发环境,这方面的内容是上机操作的基础。

2. 上机实验指导。为了方便读者上机练习,在这部分设计了 14 个实验。这些实验和课堂教学紧密配合,通过有针对性的上机实验,可以更好地熟悉 FORTRAN 90 程序设计的方法,并培养一定的应用开发能力。

3. 常用算法设计。“如何编写程序”是读者学习的难点,作者将问题进行分类,总结每一类问题的编程思路,并给出大量的程序实例。教学实践表明,这样做对提高初学者的编程能力是大有裨益的。同样这些内容对学习其他高级语言程序设计(C 程序设计、VB 程序设计等)也是有帮助的。

4. 程序调试方法。程序调试是程序设计的重要环节。编写程序时出错是难免的,问题是如何在短时间内发现并纠正程序中的错误,这就要培养较强的调试程序的能力。作者介绍程序调试的一般方法,并结合 Fortran PowerStation 4.0 集成开发环境说明如何运用这些方法。

5. 习题选解。为帮助读者进行课外练习,特意增加这部分内容。对于参加各种计算机考试的读者来说,这部分内容也是很好的辅导材料。

本书由刘卫国任主编,戴忠任副主编。戴忠编写第一章、第四章第三节、第五章第一~五节,蔡旭晖编写第二章、第五章第六节,刘卫国编写第三章、第四章第一、二节。蔡旭晖、舒卫真上机调试了部分程序。全书由刘卫国教授审阅、定稿。

由于时间仓促及水平所限,书中不当或错误之处,敬请读者批评指正。

编 者  
2003 年 1 月

# 目 录

第一章 Fortran PowerStation 4.0 集成开发环境 .....	(1)
1.1 系统安装和启动 .....	(1)
1.1.1 系统要求 .....	(1)
1.1.2 安装步骤 .....	(1)
1.1.3 Fortran PowerStation 4.0 的启动 .....	(3)
1.2 Fortran PowerStation 4.0 的用户界面 .....	(3)
1.2.1 菜单栏 .....	(3)
1.2.2 工具栏 .....	(5)
1.2.3 编辑窗口 .....	(6)
1.3 项目管理 .....	(7)
1.3.1 建立项目工作间 .....	(8)
1.3.2 给项目工作间添加文件 .....	(9)
1.3.3 给项目工作间删除文件 .....	(10)
1.4 文件编译与连编 .....	(10)
1.5 项目运行 .....	(12)
第二章 上机实验指导 .....	(13)
2.1 实验要求 .....	(13)
2.2 实验内容 .....	(14)
实验一 FORTRAN 程序设计基础 .....	(14)
实验二 顺序结构程序设计 .....	(18)
实验三 选择结构程序设计 .....	(21)
实验四 循环结构程序设计 .....	(25)
实验五 常用算法 .....	(31)
实验六 FORTRAN 数据类型 .....	(34)
实验七 格式输入输出 .....	(37)
实验八 数组 .....	(40)
实验九 子程序 .....	(45)
实验十 结构体 .....	(54)
实验十一 指针 .....	(56)

实验十二 文件 .....	(59)
实验十三 FORTRAN 图形用户界面设计 .....	(64)
实验十四 综合实验 .....	(67)
<b>第三章 常用算法设计</b> .....	<b>(76)</b>
3.1 累加与累乘问题 .....	(76)
3.2 数字问题 .....	(80)
3.3 数值计算问题 .....	(85)
3.4 数组的应用 .....	(89)
3.5 子程序的应用 .....	(93)
3.6 解不定方程 .....	(97)
思考题 .....	(98)
<b>第四章 程序测试与调试</b> .....	<b>(101)</b>
4.1 程序测试 .....	(101)
4.2 程序调试 .....	(102)
4.2.1 诊断错误的实验方法 .....	(102)
4.2.2 诊断错误的推理技术 .....	(104)
4.2.3 错误修改的原则 .....	(104)
4.3 程序调试实例 .....	(105)
4.3.1 程序的动态调试 .....	(105)
4.3.2 常用的调试语句 .....	(108)
4.3.3 系统调试工具 Debugger .....	(110)
<b>第五章 习题选解</b> .....	<b>(113)</b>
5.1 填空题 .....	(113)
5.2 阅读程序题 .....	(115)
5.3 填空编程题 .....	(128)
5.4 改错题 .....	(142)
5.5 补充编程题 .....	(156)
5.6 编程题 .....	(165)
参考文献 .....	(190)

# 第一章 Fortran PowerStation 4.0 集成开发环境

本章首先介绍 Fortran PowerStation 4.0 集成开发环境的安装与启动,然后介绍它的工作窗口及其操作。Fortran PowerStation 4.0 可以在 Windows 95/98、Windows NT 3.51 或其他更高版本的操作系统下运行。限于篇幅,本章主要介绍在 Windows 95/98 操作系统下的使用。

## 1.1 系统安装和启动

### 1.1.1 系统要求

Microsoft Fortran PowerStation 4.0 对运行环境的基本要求如下:

- 80486 或以上处理器,16M 以上内存空间;
- 足够的硬盘空间,标准版和专业版的典型安装分别需要 45MB 和 70MB 硬盘空间;
- VGA 显示器;
- Windows 95/98、Windows NT 3.51 或以上操作系统。

### 1.1.2 安装步骤

下面以 Window 95/98 操作系统为例介绍 Microsoft Fortran PowerStation 4.0 的安装过程:

(1) 将含有 Microsoft Fortran PowerStation 4.0 的光盘放入光盘驱动器,运行安装程序(Setup.exe),屏幕出现欢迎画面,如图 1.1 所示。安装向导会引导进行下面的安装。

(2) 单击 Continue 按钮,进入选择安装类型对话框,如图 1.2 所示。共有 4 种安装类型可以选择。

- 典型安装(Typical)

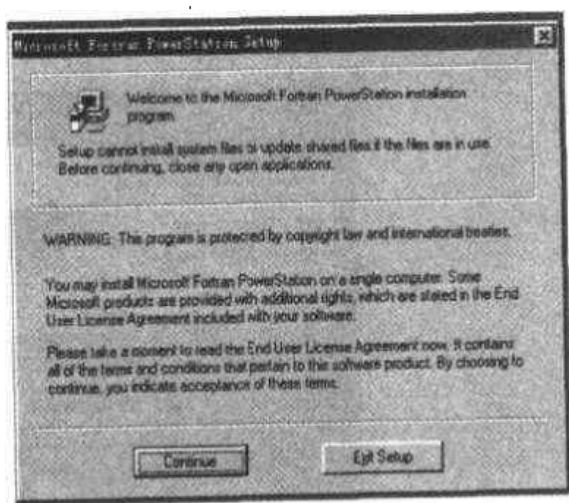


图 1.1 安装向导欢迎画面



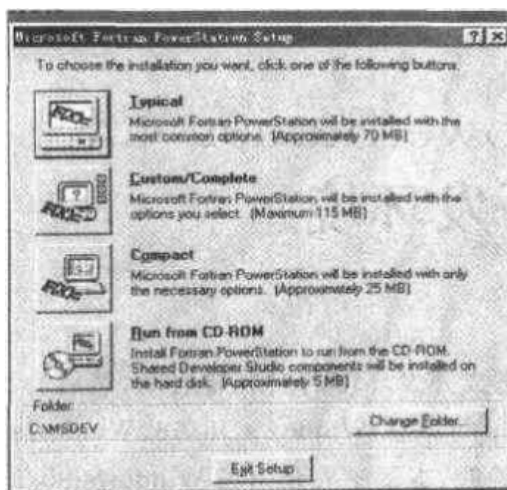


图 1.2 安装类型对话框和改变安装目录按钮



图 1.3 选择文件夹对话框

安装后将包含最常用的组件。对于大多数普通用户推荐使用这一选择。

- 用户自定义/完全安装(Custom/Complete)

此选项针对那些对 Fortran PowerStation 4.0 比较熟悉的用户。如果知道需要哪些组件,可以选择此选项,由用户自己来定义那些需要的组件。

- 压缩安装(Compact)

安装后只包含运行所必需的必要组件。如果用户的硬盘空间比较小,可以选择这一安装方式。

- 从光驱运行安装(RUN from CD-ROM)

安装后的编译和运行都需要光盘的支持。因为速度太慢,一般不使用这种方式。

(3) 单击 Change Folder 按钮,可以改变安装路径。系统默认的安装路径为 C:\MSDEV。

(4) 选择安装类型和安装路径后,进入选择文件夹对话框,如图 1.3 所示。

(5) 选择文件夹后,安装向导开始根据前面的选择将文件拷贝到指定的路径(图 1.4)。

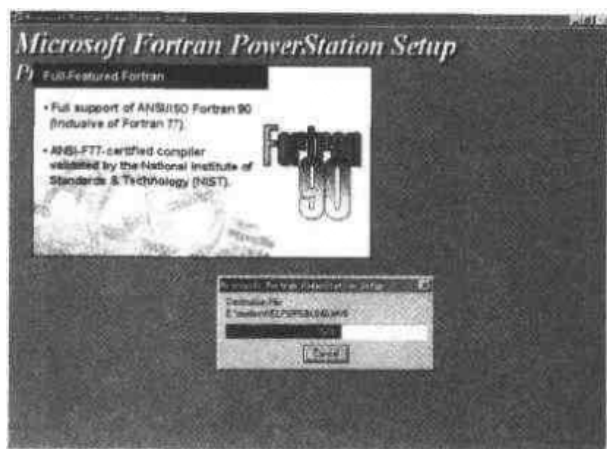


图 1.4 文件拷贝过程

(6) 拷贝完成提示修改 Autoexec. bat 文件,按 YES 完成。

(7) 最后需要重新启动计算机(图 1.5),Fortran PowerStation 4.0 的安装就完成了。

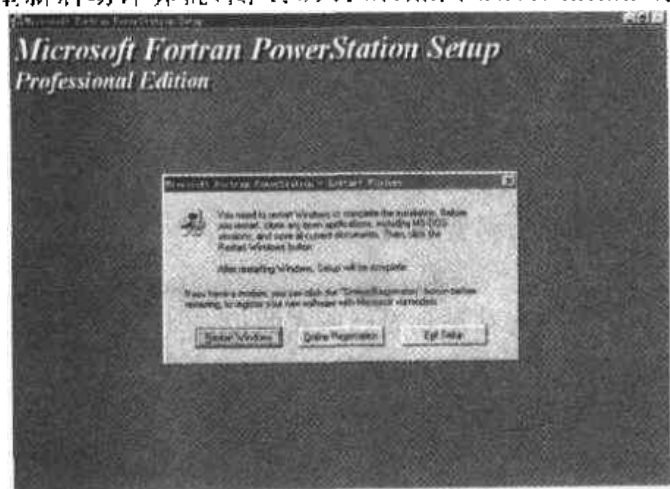


图 1.5 安装完成提示画面

### 1.1.3 Fortran PowerStation 4.0 的启动

在 Windows 95/98 操作系统下启动 Fortran PowerStation 4.0 主要有 3 种方法:

- (1) 单击“开始”菜单,在“程序”中选择 Fortran Power Station 4.0,再从中找到 Microsoft Developer Studio 选项,单击这一选项就可进入 Fortran PowerStation 4.0;
- (2) 打开资源管理器,找出 Fortran PowerStation 4.0 的安装路径,双击 Msdev. exe;
- (3) 先建立 Fortran PowerStation 4.0 的快捷方式并放在桌面上,然后在桌面上双击该快捷方式图标。

## 1.2 Fortran PowerStation 4.0 的用户界面

启动 Fortran PowerStation 4.0 后,就进入了 Fortran PowerStation 4.0 的工作窗口(如图 1.6 所示)。

### 1.2.1 菜单栏

Fortran PowerStation 4.0 工作窗口的菜单栏由 8 个菜单项组成,下面分别介绍其功能。

#### 1. File 菜单项

New 新建文件或项目

Open 打开已有的文件

Close 关闭打开的文件

Open Workspace 打开已有的

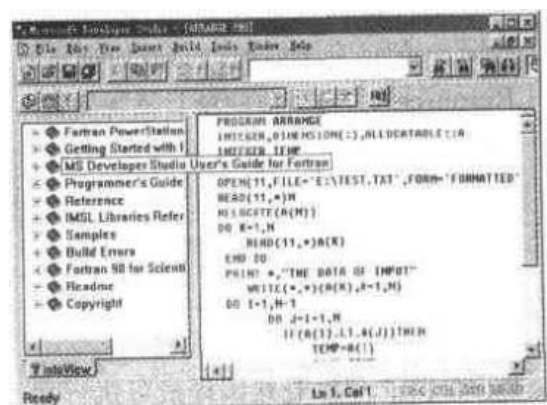


图 1.6 FORTRAN PowerStation V4.0 的工作窗口

## 工作间

Close Workspace 关闭已打开的工作间

Save 保存当前编辑窗口的内容

Save As 把当前编辑窗口的内容用另外的文件名保存

Save All 保存全部编辑窗口中的内容

Cut 把选定的内容剪切并送至剪贴板

Copy 把选定的内容复制到剪贴板

### 2. Edit 菜单项

Paste 把剪贴板中的内容粘贴到光标所在的位置

Delete 删除选定的内容

Select All 选中当前文档的全部内容

Find 查找指定的字符串

Replace 查找并替换指定的字符串

GoTo 到某一行、某一书签或某一定义

Breakpoints 设置断点

### 3. View 菜单项

FullScreen 全屏显示编辑窗口中的内容

Toolbars 设置工具栏

Project Workspace 用于恢复默认的工作间

InfoViewer Topic 用于在编辑窗口内显示信息资料主题

Output 用于显示输出信息窗口

Watch 调试时用于显示用户自定义变量列表观察窗口

Variables 调试时用于显示当前作用域范围内所有变量的值

Registers 调试时用于显示寄存器的状态

Memory 调试时用于显示内存中的状态

CallStack 调试时用于显示堆栈的调用过程

Disassembly 调试时用于显示反汇编信息窗口

### 4. Insert 菜单项

File 把已有文件的内容添加到当前文件中

Resource 为当前的项目添加资源

Resource Copy 把已有的资源拷贝到当前项目

Files into Project 把已有文件的内容添加到当前项目中

Project 在已有的项目中添加新项目

### 5. Build 菜单项

Compile 编译当前的文件

Build 把项目连接编译成可执行文件

Rebuild All 重新连接编译所有的文件

Batch Build 批处理连接编译形式

Stop Build 终止文件的连接编译

Update All Dependencies 用于更新所有的相关文件

Debug 用于设置各种调试选项

Execute 用于运行可执行文件

Settings 用于设置项目的运行环境

Configuration 用于设置项目的配置

Subprojects 用于设置子项目

#### 6. Tools 菜单项

Browse 用于浏览项目的总体信息

Close Browse InfoFile 用于关闭项目的总体信息窗口

Profile 用于记录可执行文件的全局信息

Remote Connection 用于远程连接

Customize 用于自定义工具栏等

Options 用于设置项目的环境

Record Keystrokes 用于录制键盘操作

Playback Recording 用于启用录制的键盘操作

#### 7. Window 菜单项

New Window 用于建立当前编辑窗口的副本

Split 用于分割当前窗口

Hide 用于隐藏工作间窗口、输出窗口等

Cascade 用于层叠排列所有编辑窗口和信息窗口

Tile Horizontally 用于水平排列所有编辑窗口和信息窗口

Tile Vertically 用于垂直排列所有编辑窗口和信息窗口

CloseAll 用于关闭所有编辑窗口和信息窗口

Windows 用于设置当前窗口

#### 8. Help 菜单项

Contents 用于显示帮助目录

Search 用于按照索引和询问等查找帮助信息

Keyboard 用于介绍所有命令的快捷键

Define Subset 用于自定义信息子集

Set Default Subsets 用于设置默认信息子集

Open Information Title 用于选定所打开的信息

Tip of the Day 用于显示常用的小技巧

Technical Support 用于打开技术支持目录

About Developer Studio 用于显示 Fortran PowerStation 4.0 的一些版本信息

### 1.2.2 工具栏

在 Fortran PowerStation 4.0 工作窗口的菜单栏下面是由许多图标组成的工具栏，

包括 Project 项目工具栏(如图 1.7 所示)、Standard 标准工具栏、Edit 编辑工具栏、Debug 调试工具栏、InfoViewer Contents 信息浏览器目录工具栏等。工具栏的功能与菜单对应栏目相同,把光标移至工具按钮上方,系统会提示按钮功能。使用工具栏中按钮可以使程序的编辑、运行、连接及调试等工作变得简单快捷。



图 1.7 Project 工具栏

Project 工具栏中按钮从左至右依次为:

- (1) 编译工具按钮 用于编译当前文件;
- (2) 连编工具按钮 用于连编项目中的所有文件;
- °(3) 终止工具按钮 用于终止连编项目;
- (4) 执行工具按钮 用于调试时开始运行项目;
- (5) 设置断点工具按钮 用于调试时在源文件中设置断点;
- (6) 取消断点工具按钮 用于调试时在源文件中取消断点。

### 1.2.3 编辑窗口

在了解编辑窗口之前,先讨论一下如何建立新的文本文件。建立新的文本文件可以遵循以下步骤:

- (1) 从 File 菜单中选择 New,弹出 New 对话框,如图 1.8 所示。

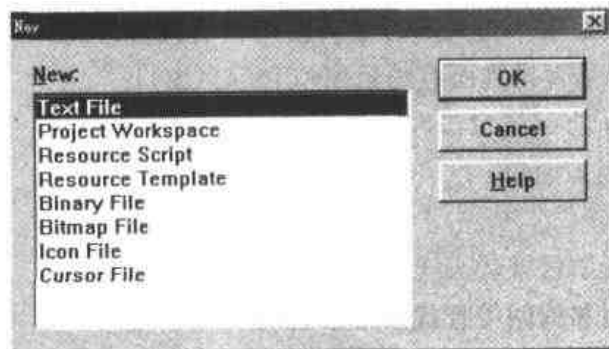


图 1.8 New 对话框

New 对话框列表中各项的作用为:

- Text File 用于建立文本文件;
- Project Workspace 用于建立项目工作间;
- Resource Script 用于建立资源脚本文件;
- Resource Template 用于建立资源模板文件;
- Binary File 用于建立二进制文件;
- Bitmap File 用于建立位图文件;
- Icon File 用于建立图标文件;

Cursor File 用于建立光标文件。

(2) 在 New 对话框的列表中选择 Text File, 按下 OK 按钮, 屏幕上出现的大片空白区域即为编辑窗口, 如图 1.9 所示。

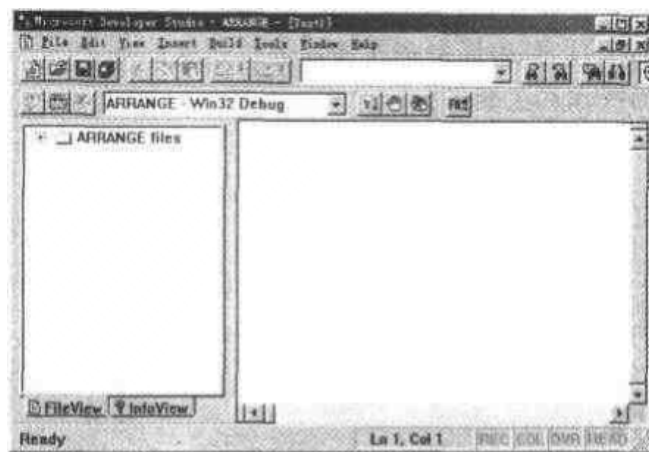


图 1.9 编辑窗口

(3) 在编辑窗口输入、编辑修改源程序, 编辑完成后可以从 File 菜单中选择 Save 或 Save As 将所编辑的文件保存到指定的目录下。

在编辑窗口内一行最多可以有 2048 个字节。源程序既可以写成自由格式, 也可以写成固定格式。固定格式的标准和传统的 FORTRAN 77 的格式相同。

如果将所编辑的源文件保存为扩展名为 .for 的文件, 则在编辑窗口内会自动添加一条宽条竖线 以方便固定格式源文件的编辑。如果将所编辑的源文件保存为扩展名为 .f90 的文件, 则编辑窗口内没有竖线。

### 1.3 项目管理

Fortran PowerStation 4.0 集成开发环境下, 在项目工作间中管理项目。项目工作间包括工作间目录的位置和目录中的一些描述工作间内容的文件。当首次建立项目工作间时, Fortran PowerStation 4.0 会自动创建一个工作间目录和一个扩展名为 .mdp 的工作间维修诊断程序文件。

创建项目工作间时生成的目录是工作间的根目录, 而在这个项目工作间之后所加入的项目都存放在这个工作间目录的子目录下。

工作间目录包含以下必需的文件:

(1) 项目工作间文件, 其扩展名为 .mdp。对于一个特定的工作间, MDP 文件包含以下一些主要内容:

- ① 对项目工作间进行实时的性能测量和管理;
- ② 程序中所设置的断点;

③ 一些关于字体和颜色等的本地设置。

(2) 项目工作间程序描述文件,其扩展名为 .mak。MAK 文件包含一些连编项目所需的信息:

- ① 项目中源文件的名字和位置;
- ② 连编项目所需的工具,如编译器和连接器等的设置;
- ③ 连编项目所需的操作和工具等。

同时,根据项目类型和设置的不同,还会产生一些临时文件。

### 1.3.1 建立项目工作间

当建立项目工作间并进行初始化以后,初始化的项目有 Debug 和 Release 两种配置。

在建立项目工作间前,有 3 种基本的方案可以选择,每一种方案按照不同的方式组织项目工作间。

#### 1. 唯一的最高层的项目

这种组织方案适合于开发单一的应用程序,这种应用程序不与其他的应用程序相关联。如果要开发单一的 QuickWin 应用程序、单一的控制台应用程序或静态链接库等可以选用这种方案。

#### 2. 带有单一子项目的最高层的项目

这种组织方案适合于开发与其他应用程序相关的应用程序。比如使用静态链接库的控制台应用程序。

#### 3. 带有多个子项目的最高层的项目

这种组织方案适合于开发一组相关的应用程序。比如要开发两个控制台程序,其中一个使用静态链接库。

下面主要介绍第一种方案。要建立一个唯一的最高层的项目工作间,可按照以下步骤进行:

(1) 从 File 菜单中选择 New,弹出 New 对话框。

(2) 在 New 对话框的列表中选择 Project Workspace,按下 OK 按钮,出现 New Project Workspace 对话框,如图 1.10 所示。

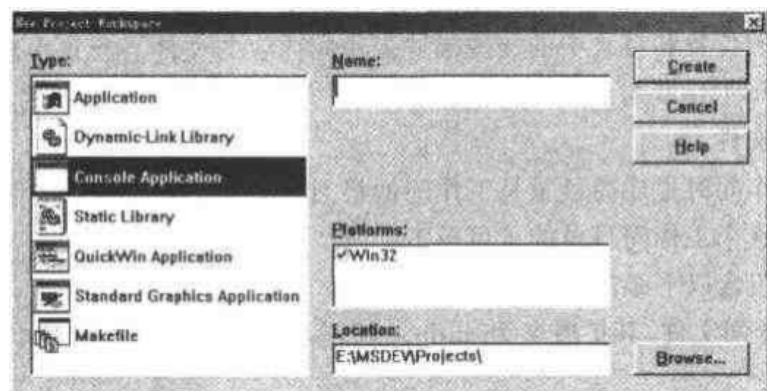


图 1.10 NewProjectWorkspacc 对话框

(3) 从类型列表中选择要创建的项目工作间的类型。其中一些项目类型的作用为:

- Application (Windows 应用程序)。它是用 Windows NT Win32 API 函数开发的图形界面应用程序。文件的扩展名为 .exe。

- Dynamic-Link Library (动态连接库)。独立于应用程序,在应用程序执行时才被编译和连接到应用程序的函数库。文件的扩展名为 .dll。

- Console Application (控制台应用程序)。它可以在控制台窗口内提供字符模式的支持,但不支持图形。文件的扩展名为 .exe。

- Static Library (静态库)。它是用属于项目的目标文件和其他库文件通过连编建立的静态库。文件的扩展名为 .lib。

- QuickWin Application (QuickWin 应用程序)。它在单个或多个窗口内支持图形输出和其他的屏幕操作。文件的扩展名为 .exe。

- Standard Graphics Application (标准图形应用程序)。它只在单个窗口内支持图形输出和各种屏幕操作。文件的扩展名为 .exe。

为了更容易理解,本书中的程序主要选择 Console Application 类型。

(4) 在名字文本框中添加所创建项目工作间名称,从平台列表中选择平台类型,在 Location 下面的文本框内添加项目工作间根目录的位置或单击 Browse 按钮来选择项目工作间根目录的位置。最后选择 Create 按钮,完成项目工作间的创建。

### 1.3.2 给项目工作间添加文件

给项目工作间添加文件是把文件加入到所有的项目工作间配置中。如按上一小节在目录 E:\MSDEV\PROJECTS 下建立项目工作间 myproject。完成之后,出现如图 1.11 所示的画面。

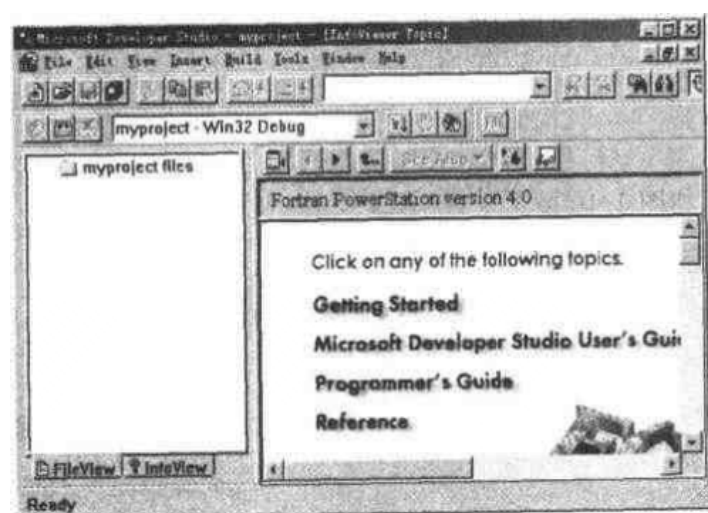


图 1.11 创建项目工作间 myproject

如果要添加新建的文件,则先建立新文件如 myfile,并将其保存在目录 E:\MSDEV\PROJECTS\myproject 下,取名为 myfile.f90。

如果要添加已有的文件,则建立新文件的过程可以省略。然后可以按照下面的步骤



为项目工作间添加文件:从 Insert 菜单中选择 Files into Project, 出现 Insert Files into Project 对话框(如图 1.12 所示),如果是已经存在的文件可以直接从文件列表中选择文件名,然后选择 Add 按钮。这样为项目工作间添加文件就完成了。

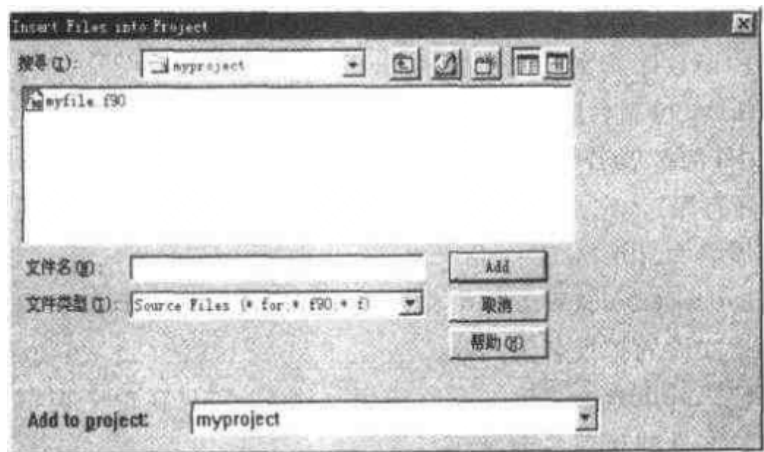


图 1.12 给项目工作间添加文件

### 1.3.3 给项目工作间删除文件

可以按照下面的步骤来为项目工作间删除文件:

(1) 在 FileView 窗口内选择要删除的文件,如图 1.13 所示。

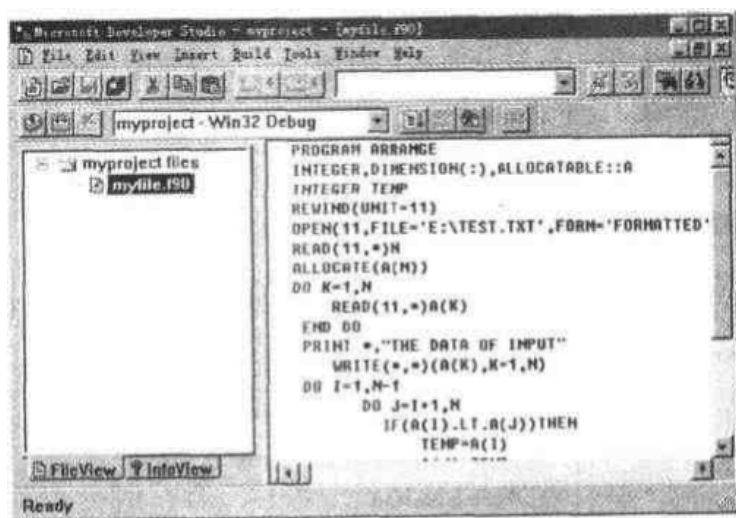


图 1.13 为项目删除文件

(2) 从 Edit 菜单中选择 Delete,这样就完成了对项目工作间文件的删除。

## 1.4 文件编译与连编

建立项目工作间并为项目添加文件后,就可以对项目工作间文件进行编译和连编。

只有经过编译和连编之后,才可以生成可执行文件。

### 1. 编译

从 Build 菜单选择 Compile 项,在编译完成之后,可以在 Output 窗口(如图 1.14 所示)中查看编译信息,其中包括项目工作间文件的错误信息或警告信息。

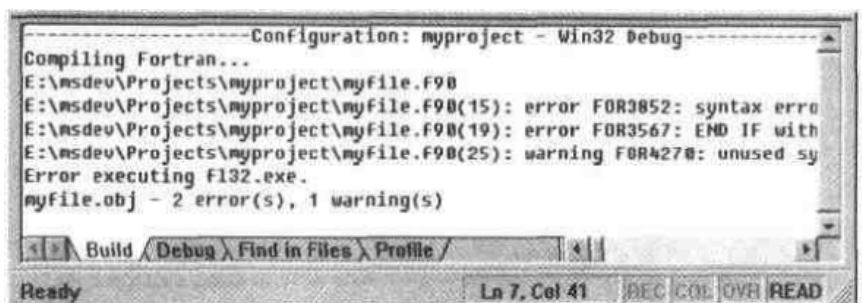


图 1.14 编译 Output 窗口

如果出现错误或警告信息,在 Output 窗口内会给出错误或警告的类型和它们在项目工作间文件中的行号。根据这些信息可以到编辑窗口中修改项目工作间文件。

可以在 Output 窗口内双击错误或警告信息所在的行,则光标会自动移到项目工作间文件出现错误或被警告的行。

### 2. 连编

从 Build 菜单选择 Build 项,在连编完成之后,同样可以在 Output 窗口中查看连编信息(图 1.15),其中包括项目工作间文件连编的错误信息。

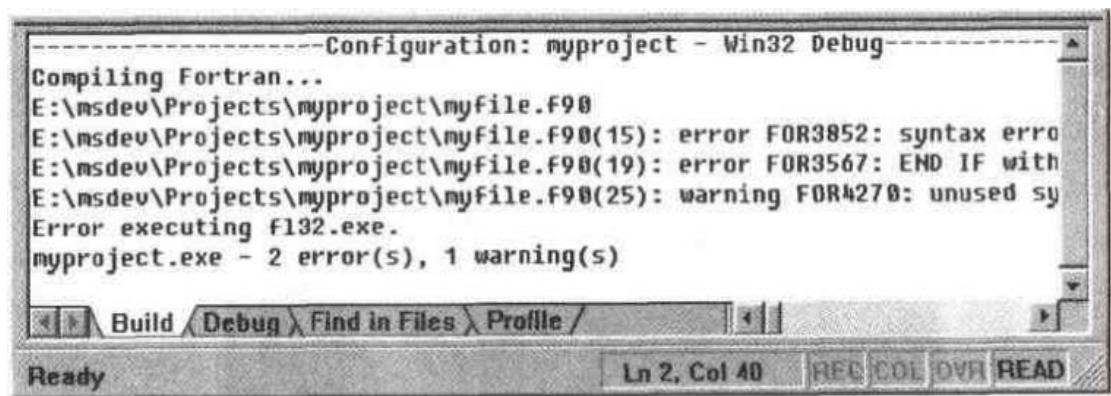


图 1.15 连编 Output 窗口

在项目的编译和连编过程中有许多选项可以设置,这些选项的设置设置在 Build 菜单下的 Settings 中。单击 Settings,出现 Project Settings 对话框,如图 1.16 所示。在该对话框中,可以根据不同程序的需要,设置不同的编译和连编选项,来对程序进行调试和优化,充分发挥系统的特性。

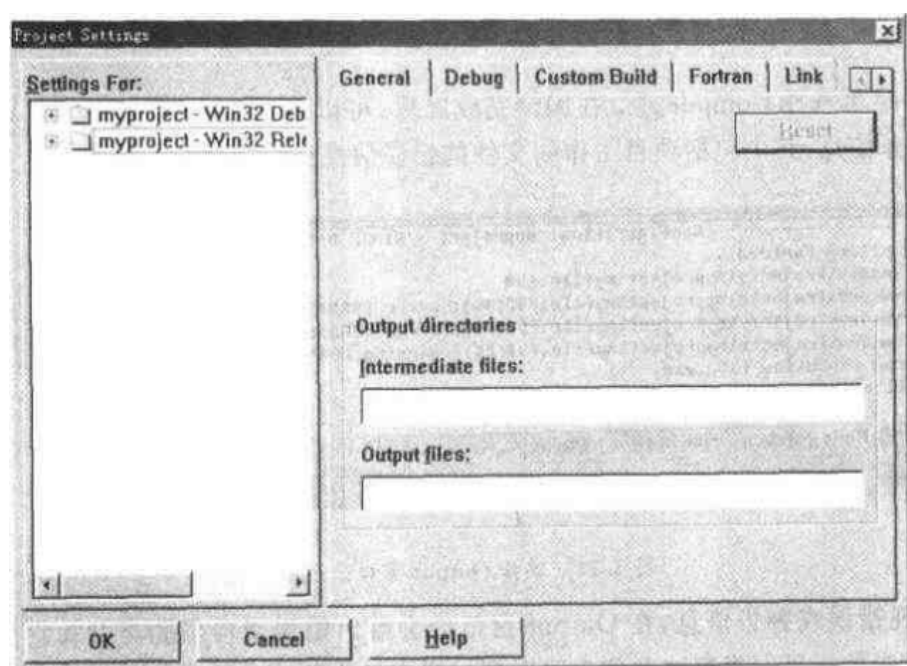


图 1.16 Project Settings 对话框

## 1.5 项目运行

如果项目通过了编译和连编,就可以运行了。要运行一个项目,可以在 Build 菜单中选择 Execute。运行前面建立的项目,输出结果如图 1.17 所示。



图 1.17 输出结果

## 第二章 上机实验指导

### 2.1 实验要求

学习程序设计,上机实验是十分重要的环节。为了方便读者上机练习,本章设计了14个实验。这些实验和课堂教学紧密配合,通过有针对性的上机实验,可以更好地熟悉FORTRAN 90的功能,掌握FORTRAN 90程序设计的方法,并培养一定的应用开发能力。前13个实验每个实验安排两个机时左右。读者也可以根据实际情况从每个实验中选择部分内容上机练习。第14个实验给出了两个综合性的问题,可以作为FORTRAN 90程序设计实践或课程设计的内容。

为了达到理想的实验效果,读者务必做到:

(1) 实验前认真准备,要根据实验目的和实验内容,复习好实验中可能要用到的命令、语句,想好编程的思路,做到胸有成竹,提高上机效率。

(2) 实验过程中积极思考,要深入分析、程序的执行结果以及各种屏幕信息的含义、出现的原因并提出解决办法。

(3) 实验后认真总结,要总结本次实验有哪些收获,还存在哪些问题,并写出实验报告。实验报告应包括实验目的、实验内容、流程图、程序清单、运行结果以及实验的收获与体会等内容。

程序设计和应用开发能力的提高需要不断的上机实践和长期的积累,在上机过程中会碰到各种各样的问题,分析问题和解决问题的过程就是经验积累的过程。只要读者按照上面3点要求去做,在学完本课程后就一定会有很大的收获,计算机应用能力就会有很大提高。

## 2.2 实验内容

### 实验一 FORTRAN 语言程序设计基础

#### 一、实验目的

1. 熟悉 Fortran PowerStation 的集成开发环境。
2. 掌握 FORTRAN 程序的上机步骤。
3. 掌握 FORTRAN 程序的结构、书写规则。

#### 二、预备知识

1. 下列叙述中正确的是：  
A) 完整的 FORTRAN 程序的第一个语句必须是 PROGRAM 语句  
B) 完整的 FORTRAN 程序中只能有一个 END 语句  
C) FORTRAN 的任何可执行语句都可以有标号  
D) FORTRAN 的所有语句都是可执行语句  
E) FORTRAN 90 的语句行可以有任意多个字符  
F) 注释语句以 & 号为标志
2. 下列叙述中不正确的是：  
A) FORTRAN 子程序可以单独编译  
B) 对一个 FORTRAN 源程序进行编译和连接无误后可生成可执行文件  
C) 即使编译和连接都正确无误, FORTRAN 程序运行时仍可能出错  
D) FORTRAN 连接的主要任务是把函数库中的函数翻译成机器指令
3. 求  $y$  的值, 请用 N-S 图表示算法。

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

4. 计算  $s = 1 + 2 + 2^2 + \dots + 2^{15}$ , 请用 N-S 图表示算法

#### 三、实验内容

1. 下列程序的功能为计算扇形面积, 其中  $T$  为扇形角度,  $R$  为半径,  $S$  为面积, 试输入数据并运行下列程序。

```
PROGRAM SQUARE
PARAMETER (PI=3.1415926)      ! 说明符号常量
OPEN(9,FILE='mydata.DAT')    ! 打开用于保存结果的文件
WRITE(*,*)"请输入 r 和 t 的值:"
READ*,r,t
S=t/360*PI*r**2
PRINT *, 'r=',r,'t=',t,'s=',s    ! 在屏幕上显示结果
WRITE(9,*) 'r=',r,'t=',t,'s=',s  ! 将结果保存至文件中
END
```

操作步骤:

(1) 启动 Fortran PowerStation 4.0 的 Microsoft Developer Studio, 进入 Fortran PowerStation 4.0 的集成开发环境。

(2) 从 File 菜单中选择 New 菜单项, 在弹出的对话框中选取 Project Workspace 建立一个项目工作间, 在接着弹出的对话框的 Name 栏内填入项目名称(如 Mypro), 单击 Create 按钮。

(3) 在 Standard 工具栏中选择新建按钮, 输入以上程序。输入完毕, 以 myfile1 为主文件名保存源程序。

(4) 选择 Insert 菜单中的 Files into Project, 向项目工作间内添加该文件。

(5) 按 Shift + F8, 对项目进行编译和连接, 如果有错误, 根据提示修改错误; 编译无错误, 则会生成扩展名为 .exe 的文件。

(6) 按 Ctrl+F5, 程序开始运行, 屏幕提示输入 r, t 的值, 这时输入:

12 60 ✓

屏幕显示运行结果。

(7) 查看数据文件。在 Standard 工具栏中单击 Open 按钮, 在弹出的对话框的“文件类型”栏内选择“所有文件(\*.\*)”, 在列出的文件中选择程序中设定的数据文件 mydata.dat, 查看结果。该结果和屏幕上显示的结果相同。

2. 输入下列程序, 学习字符型数据的输入方法。

```
PROGRAM exercise
CHARACTER word * 20
OPEN(10,FILE='english.DAT')
WRITE(*,*)"请输入一个英文单词:"
READ*,word
PRINT *, word
WRITE(10,*) word
END
```

操作步骤:

(1) 新建一个文件, 输入上述程序, 以 myfile2 为主文件名保存此程序。

(2) 在项目工作间 mypro 中删除第 1 题的源程序文件 myfile1, 将 myfile2 加入项目

工作间。

(3) 按 Shift + F8, 对项目进行编译和连接。

(4) 按 Ctrl + F5, 程序开始运行, 按提示输入一个英文单词, 注意输入单词时要带引号。例如:

"FORTRAN" ↵

3. 输入下列程序, 观察程序运行结果。

```
READ *, n
DO i=1, n
    WRITE(*, 100) ("*", j=1, i)
END DO
100 FORMAT(1X, 20A1)
END
```

实验要求:

(1) 运行程序, 输入一个小于 20 的整数, 观察运行结果。

(2) 将第 3 条语句改为: WRITE(\*, 100) ("\*", j=1, n)

重新编译连接后, 运行程序, 观察结果。

4. 已知任意五边形的五条边长分别为 a, b, c, d, e, 从一个顶点出发的两条对角线的长度分别为 m, n, 编写程序求五边形的面积。

分析: 把五边形分成三个三角形, 然后把面积相加即可得到所求。按照如下公式分别计算三角形的面积:  $\text{area} = \sqrt{t(t-a)(t-b)(t-c)}$ , 其中  $t = \frac{a+b+c}{2}$ , a, b, c 为三角形的边长。

```
IMPLICIT NONE
REAL a, b, c, d, e, m, n, area, s
PRINT *, "请输入 a, b, c, d, e, m, n 的值:"
READ *, a, b, c, d, e, m, n
s = area(a, b, m) + area(m, c, n) + area(n, d, e)
PRINT *, "五边形的面积为:", s
END
```

操作步骤:

(1) 新建一个文件, 输入上述程序, 以 myfile4 为主文件名保存此程序。

(2) 新建一个文件, 输入下列子程序, 以 sub4 为主文件名保存此程序。

```
FUNCTION area(a, b, c)
IMPLICIT NONE
REAL a, b, c, t, area
IF (a + b > c .AND. ABS(a - b) < c) THEN
    t = (a + b + c) / 2
    area = SQRT(t * (t - a) * (t - b) * (t - c))
END FUNCTION
```

```
ELSE  
    STOP "输入数据错误!"  
END IF  
END
```

(3) 在项目工作间 mypro 中删除原来的源程序文件,将 myfile3、sub4 加入项目工作间。

(4) 按 Shift + F8,对项目进行编译和连接。

(5) 按 Ctrl+F5,程序开始运行,按提示输入数据。

要求分别输入以下两组数据,观察运行结果:

3 3 3 3 3 4 5 ✓

3 3 3 3 3 1 1 ✓



## 实验二 顺序结构程序设计

### 一、实验目的

1. 掌握 FORTRAN 90 中基本数据类型及其描述、运算。

- (1) 常量和变量的概念；
- (2) 定义变量类型的方法；
- (3) 符号常量及其定义方法(PARAMETER 语句)；
- (4) 运算符及运算优先级；
- (5) 算术表达式和字符表达式及其求值规则；
- (6) 不同类型数据的混合运算。

2. 最基本的语句。

- (1) 赋值语句；
- (2) 表控格式的输入和输出语句；
- (3) 程序执行控制语句。

### 二、预备知识

1. 指出下列哪些是合法的变量名：

110      \$co      5ac      ex-1      a\_1      c.d      e3      true

2. 指出下列常数中哪些符合 FORTRAN 90 语法规则：

1.e3    019    (2+3)e(4-2)    4e3.2    .19

3. 判断下列说法中哪些是正确的,哪些是错误的：

- (1) PARAMETER(N=21)定义了一个符号常量 N,其值在程序中可以改变。
- (2) 在 FORTRAN 程序中,各类语句的位置是任意的。
- (3) cos 不可以作为变量名。
- (4) 在 FORTRAN 中“=”表示相等的意思,如“A=B”,表示变量 A 与变量 B 的值是相等的。

4. 用 FORTRAN 表达式表示下列各题：

- |  |  |
|--|--|
| (1) $ ab  + 2x^2 + e^x$                  | (2) $\sin^4 x + \ln y$                 |
| (3) $\arctan\left(\frac{xy}{x+y}\right)$ | (4) $\frac{-b + \sqrt{b^2 - 4ac}}{2a}$ |

(5) 求实数  $x$  的小数部分

(6) 将实数  $x$  保留两位小数,第三位小数进行四舍五入处理

5. 修改下列程序段中的错误：

(1) INTEGER x,y,z,aver

x=7

y=5

z=5

aver=(x+y+z)/3

(3) PARAMETER (G=0.981)

G=G\*10

(2) a = 3.45

b = c = a

(4) 求 65° 的余弦值

y = cos(65)

6. 分析以下程序,写出运行结果。

(1) IMPLICIT none

INTEGER m,n

REAL a,b,x

m=7

n=4

a=38.4

b=6.4

x=m/2+n\*a/b+1/2

PRINT \*,x

END

(2) IMPLICIT none

INTEGER m,n

REAL a,b,x

m=7

n=4

a=-38.4

b=6.4

x=mod(m,n)+sign(a,b)

PRINT \*,x

END

### 三、实验内容

1. 输入下列程序。

IMPLICIT NONE

INTEGER i,x

PRINT \*, "请输入一个整数:"

READ \*,x

DO i=2,x-1

IF (MOD(x,i) == 0) STOP "合数"

END DO

PRINT \*, "素数"

END

要求分别输入 27 和 29,观察运行结果。

2. 编写程序将华氏温度转化为摄氏温度,公式为:  $C = \frac{5}{9}(F-32)$ 。

要求:输入一个华氏温度值 F,按照公式将其转化为摄氏温度值 C。

3. 编写程序,其功能是:输入 a,b,x 的值,计算并输出 y 的值。

$$y = \frac{e^{ax} + \ln^2 x}{5b}$$

当  $x=1.234, a=-5, b=0.9876$  时,  $y = \underline{\hspace{2cm}}$ 。

4. 自由落体位移公式为  $s = \frac{1}{2}gt^2 + v_0t$ , 其中  $v_0$  为初始速度,  $g$  为重力加速度,  $t$  为经历的时间。编写程序求位移量  $s$ 。设  $v_0 = 4.8\text{m} \cdot \text{s}^{-1}$ ,  $t = 0.5\text{s}$ ,  $g = 9.81\text{m} \cdot \text{s}^{-2}$ , 程序中将  $g$  定义为符号常量,  $v_0$  和  $t$  的值从键盘输入。

5. 编写一个程序, 其功能是: 从键盘输入一个三位整数, 计算并输出该数各位数字之和。

6. 分期付款的计算。设从银行借了一笔款  $d$ , 已知准备每月偿还的款额  $p$  和月利率  $r$ , 计算需多少个月才能还清。计算还清此款所需月数的公式为:

$$m = \frac{\lg p - \lg(p - dr)}{\lg(1 + r)}$$

7. 一个笼中有鸡、兔若干只, 共有  $m$  个头,  $n$  只脚, 编程求当  $m = 30$ ,  $n = 90$  时, 笼中鸡、兔各有几只。

8. 以下程序不用第三个变量, 实现将两个数进行值的互换操作。请选择填空。

```
IMPLICIT none
```

```
INTEGER a,b
```

```
READ *,a,b
```

```
a= (1)
```

```
b= (2)
```

```
a= (3)
```

```
PRINT *,a,b
```

```
END
```

(1) A)  $a+b$       B)  $a-b$       C)  $a*b$       D)  $a/b$

(2) A)  $a+b$       B)  $a-b$       C)  $b-a$       D)  $a*b$

(3) A)  $a+b$       B)  $a-b$       C)  $a*b$       D)  $a/b$

## 实验三 选择结构程序设计

### 一、实验目的

1. 掌握逻辑表达式的书写方法和逻辑型数据的使用。
2. 学会正确使用逻辑运算符和逻辑表达式。
3. 熟练掌握块 IF 结构、块 CASE 结构、逻辑 IF 语句和算术 IF 语句。

### 二、预习知识

1. 下列说法中正确的是：  
A) 块 IF 结构中的 THEN 块或 ELSE 块不能是空块  
B) 块 IF 结构中至少有一个 ELSE IF 语句或 ELSE 语句  
C) 每一个 ELSE IF 语句要有一个对应的 ENDIF 语句  
D) 一个块 IF 结构中只能有一个 ENDIF 语句
2. 假设  $A = .TRUE.$ ,  $B = .FALSE.$ ,  $C = .FALSE.$ , 则  $(A .OR. B) .AND. C$  与  $A .OR. .NOT. B$  的值分别是：  
A)  $.FALSE.$  与  $.TRUE.$                       B)  $.TRUE.$  与  $.FALSE.$   
C)  $.TRUE.$  与  $.TRUE.$                       D)  $.FALSE.$  与  $.FALSE.$
3. 用 FORTRAN 表达式表示下列各题：  
(1) A 和 B 之一为 0, 但不全为 0。  
(2) I 是十位数为 6 的三位自然数(如 365)。  
(3) 角 T 在第 - 或第三象限。  
(4)  $|x_1 - x_2| \leq 10^{-6}$ 。
4. 阅读下列程序, 写出下列程序的运行结果：  
(1)  

```
READ(*,*) N
X=1.0
IF (N, GE, 0) X=2 * X
IF (N, GE, 5) X=2 * X+1.0
IF (N, GT, 15) X=3 * X-1.0
WRITE(*,*) X
END
```

运行上述程序时, 如果从键盘输入 15  $\swarrow$ , 则最后输出的 X 值为:  
A) 14.0      B) 2.0      C) 1.0      D) 5.0

(2)

```
IMPLICIT none
LOGICAL no1,no2
REAL a,b,c,d,x
a=2.5;b=7.5;c=5.0;d=6.0;x=0
no1=.TRUE.
no2=.FALSE.
IF(.NOT.no1.OR.c.EQ.d.AND.no2) x=x+1
IF(a+b/2.0.NE.c-d.OR.c.NE.d) x=x+1
PRINT *,x
END
```

运行程序,输出的 X 值为: \_\_\_\_\_。

(3)

```
LOGICAL P
READ(*,*)X,Y
P=.TRUE.
IF(X+Y.LT.X) P=.FALSE.
Z=Y
IF(P) THEN
    Z=X+Z
ELSE
    Z=X
END IF
```

```
WRITE(*,*)Z
```

```
END
```

运行时输入 3.6, -1.2 ✓, 输出的 Z 值是:

A) 3.6      B) 2.4      C) 0.0      D) -1.2

(4)

```
LOGICAL X,Y
READ(*,100) X,Y
100 FORMAT(2L4)
IF (X) THEN
    P=1.0
ELSE IF (Y) THEN
    P=2.0
ELSE IF (.NOT.X.OR.Y) THEN
    P=3.0
ELSE
    P=0.0
```

```

END IF
WRITE( *, * ) P
END

```

运行上述程序时,如果从键盘输入 FXYZTABC ✓,则最后输出的 P 的值为:

A) 2.0      B) 3.0      C) 1.0      D) 0.0

### 三、实验内容

1. 输入年月,求该月的天数。

用 year、month 分别表示年、月,day 表示每月的天数。注意两点:

(1) 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;

(2) 年份能被 4 整除,但不能被 100 整除,或者能被 400 整除的年均是闰年。

完善下列程序:

```

IMPLICIT none
INTEGER year,month,day
READ *,year,month
SELECT CASE(month)
  CASE (2)
    IF(     (1)     ) THEN
      day=29
    ELSE
      day=28
    END IF
  CASE (     (2)     )
    day=30
  CASE DEFAULT
    day=31
END SELECT
PRINT 100,year,"年",month,"月有",day,"天"
100 FORMAT(1X,I4,A2,I2,A4,I2,A2)
END

```

2. 以下程序为输入一个数 I,判断其是否是水仙花数。所谓水仙花数是指一个三位整数各位数字之立方和等于该数本身,例如  $153 = 1^3 + 5^3 + 3^3$ ,所以 153 是水仙花数。完善程序。

```

READ *,I
I1 =     (1)          ! 求该数的百位数字
I2 =     (2)          ! 求该数的十位数字
I3 =     (3)          ! 求该数的个位数字

```

```
M=I1**3+I2**3+I3**3      ! 求各位数字的立方和
```

```
IF(M.EQ.1) PRINT *,1,' is a daffodil number.'
```

```
END
```

3. 由键盘输入三条线段的长度,判断这三条线段能否构成三角形,如能构成三角形,则计算三角形的面积。

4. 输入一个自然数,若为奇数则输出其平方根,否则输出其立方根。

5. 编写程序计算下列分段函数:

$$y = \begin{cases} \sqrt{4 + \ln x} & 1 \leq x < 2 \\ x^2 + e^x & 2 \leq x < 3 \\ \lg x & 3 \leq x < 4 \\ 1 + x + x^2 & \text{其他} \end{cases}$$

设计测试数据对程序进行检测,应注意对分段函数的每一区间都测试到,尤其对每一区间的端点(如  $x=1, 2, 3, 4$ )进行测试,以防设计中出现的疏忽。

6. 某商场对顾客所购买的商品实行打折销售,标准如下(商品价格用 price 来表示):

price < 200            没有折扣

200 ≤ price < 500    3%折扣

500 ≤ price < 1000   5%折扣

1000 ≤ price < 5000   8%折扣

5000 ≤ price          10%折扣

求所售商品的实际价格。

7. 由键盘输入任意四个实数 a, b, c, d, 按由小到大的顺序把它们打印出来。

8. 某航空公司规定:在旅游旺季 7~10 月,如果订票超过 20 张,优惠票价的 15%;订票 20 张以下,优惠票价的 5%;在旅游淡季 1~4 月、11 月,如果订票超过 20 张,优惠票价的 30%;订票 20 张以下,优惠票价的 20%。编写程序,根据月份和旅客订票张数决定优惠率。

9. 满足下列条件之一,即获得学习优良奖:

(1) 所考 5 门课成绩总分超过 450 分;

(2) 每门课都在 88 分以上;

(3) 前 3 门课的成绩都在 95 分以上,其他两门课的成绩都在 80 分以上。

输入某学生 5 门课的成绩,判断是否符合优良奖的条件。

## 实验四 循环结构程序设计

### 一、实验目的

1. 掌握用 DO 语句和 DO WHILE 语句实现循环的方法。
2. 掌握循环结构的实现方法。
3. 掌握在程序设计中用循环的方法实现各种算法(如穷举法、迭代、递推等)。

### 二、预备知识

#### 1. 阅读程序:

```

READ( *, *) A,B
P=MOD(B,A)
DO WHILE(ABS(P).GT.0.1)
    B=A
    A=P
    P=MOD(B,A)
END DO
WRITE( *, *) A
END

```

运行上述程序时,如果从键盘输入 3.6,2.4 $\swarrow$ ,则最后输出的 A 值为:

- A) 3.6                      B) 2.4                      C) 1.2                      D) 0.0

#### 2. 阅读下列程序:

```

DO I=1, 5
    WRITE( *, *) (' ',J=1, 5-I),(' # ',K=1, 2*I-1)
END DO
END

```

程序运行后,输出的图形为(□表示空格):

A) □□□□#

□□□##

□□####

□#####

#####

B) #####

#####

#####

#####

#####



C)  $\square\square\square\square\#\#\#\#\#$   
 $\square\square\square\#\#\#\#\#\#$   
 $\square\square\#\#\#\#\#\#\#$   
 $\square\#\#\#\#\#\#\#\#$   
 $\#\#\#\#\#\#\#\#\#$

D)  $\square\square\#$   
 $\square\#\#\#$   
 $\#\#\#\#\#$   
 $\square\#\#\#$   
 $\square\square\#$

3. 阅读下列程序:

```
N=0
DO K=1,3
  N=N+K
END DO
WRITE(*,*) N,K
END
```

运行上述程序后,输出的结果为:

A) 3      3      B) 3      4      C) 6      3      D) 6      4

4. 阅读下列程序:

```
N=0
DO K=0,1
  DO J=0,2
    N=N+1
  END DO
  N=N-J
END DO
M=K+J
WRITE(*,*) N,M
END
```

运行上述程序后,输出的结果为:

A) 0      3      B) 0      5      C) 1      3      D) 1      5

5. 阅读下列程序:

```
N=0
DO K=3,5
  DO J=2,K
    IF(MOD(K,J)==0) EXIT
    N=N+J
  END DO
END DO
WRITE(*,*) N
END
```

运行上述程序后,输出的 N 的值为:

A) 2                      B) 9                      C) 11                      D) 16

6. 阅读下列程序段:

```
S=0.0  
DO J=0,N  
    S=S+X  
    X=X+0.5  
END DO
```

下面程序段中与上述程序段等价的是:

A) S=X  
 J=0  
 DO WHILE(J. LT. N)  
 X=X+0.5  
 S=S+X  
 J=J+1  
 END DO

B) S=0.0  
 J=0  
 DO WHILE(J. LE. N)  
 S=S+X  
 J=J+1  
 X=X+0.5  
 END DO

C) S=0.0  
 J=0  
 DO WHILE (J. LE. N)  
 X=X+0.5  
 J=J+1  
 S=S+X  
 END DO

D) S=X  
 J=0  
 DO WHILE(J. LT. N)  
 S=S+X  
 X=X+0.5  
 J=J+1  
 END DO

7. 下面程序段中不是死循环的有哪些?

(1) N=1

```

DO WHILE(N>0)
    N=N+1
END DO
(2) N=1
DO WHILE(N<20)
    IF(MOD(N,2)==0)N=N+1
END DO
(3) LOGICAL::LOG1=.TRUE.
    N=1
DO WHILE(LOG1)
    N=N+1
    IF(N==20) EXIT
END DO
(4) N=100
DO WHILE(N<=100)
    N=N-1
END DO

```

### 三、实验内容

1. 回文数是指正读和反读都一样的正整数,例如 66,121 等,求[10,500]以内回文数的数目 K。完善下列程序。

```

K= (1)
DO I=10,500
    IF (2) THEN
        I1=I/10
    ELSE
        I1=I/100
    ENDIF
    I2=MOD(I,10)
    IF(I1.EQ.I2) K= (3)
END DO
PRINT *, '1~500 以内回文数的个数为:',K
END

```

2. 求 100~1000 之间的所有素数,要求输出这些素数的个数和平均值。其中 COUNT、SUM 为这些素数的个数、和。

```

INTEGER COUNT
COUNT=0
SUM= (1)

```

```

DO N= (2)
  J=SQRT(N)
  I=2
  DO WHILE( (3) )
    I=I+1
  END DO
  IF(I.GT.J)THEN
    COUNT=COUNT+1
    SUM= (4)
  ENDIF
END DO
WRITE(*,100)COUNT, (5)
100 FORMAT(1X,'COUNT=',I3,5X,'AVERAGE=',F12.6)
END

```

3. 求  $Y = \sum_{n=1}^m (S_n + \sum_{i=1}^n i)$ , 其中  $S_n = \begin{cases} n^2 & n \text{ 为奇数} \\ 2n & n \text{ 为偶数} \end{cases}$

当  $m=10$  时,  $Y$  的值是多少?

```

IMPLICIT NONE
INTEGER I,M,N
REAL::Y=0,S,F
READ*,M
DO N=1,M
  (1)
  DO I=1,N
    S=S+I
  END DO
  IF( (2) ) THEN
    F=N*N
  ELSE
    F=2*N
  END IF
  Y= (3)
END DO
PRINT*,"Y=",Y
END

```

4. 编写程序统计 100 到 300 之间能同时被 3 和 7 整除的数的个数  $N$ , 计算并输出这些数的平方和。

5. 1995 年我国有 12 亿 3 千万人口。以年人口平均增长率为 1.15% 计算, 从 1995

年起经过多少年我国人口将会增长到 15 亿?

6. 设  $s = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}$ , 求与 8 最接近的  $s$  的值及与之对应的  $n$  值。

7. 输入  $x$ , 按下式计算  $\arctan x$  的近似值:

$$\arctan x = x - \frac{1}{3}x^3 + \frac{1}{5}x^5 - \frac{1}{7}x^7 + \cdots + \frac{(-1)^n}{2n+1}x^{2n+1}$$

8. 计算  $s = x + \frac{x^5}{5!} + \frac{x^9}{9!} + \cdots + \frac{x^{4n+1}}{(4n+1)!}$ 。

9. 所谓“完数”是指一个数除自身外的所有因子之和恰好等于这个数, 如 28, 除自身外的因子有 1, 2, 4, 7, 14, 显然  $28 = 1 + 2 + 4 + 7 + 14$ , 因此 28 是一个完数。编写程序, 找出 1~1000 之间的所有完数, 输出第 3 个完数, 计算所有完数的和。

10. 求  $[1, 100]$  间有奇数个不同因子的整数, 共多少个, 其中最大的一个。

11. 设  $abc \times e = dcba$ , ( $a$  非 0,  $e$  非 0 非 1)。求满足条件的  $abcd$  与  $e$ 。

12. 计算  $F_{ij}$  和  $S$ :

$$F_{ij} = \frac{\sin(X_i + Y_j)}{1 + X_i Y_j}, S = \sum_{i=1}^5 \sum_{j=1}^{10} F_{ij}$$

$$X_i = 1, 3, 5, 7, 9$$

$$Y_j = 2.1, 2.2, 2.3, \dots, 3.0$$

13. 把一张一元的钞票换成一分、二分和五分硬币, 每种至少一枚, 问有多少种方案。

14. 爱因斯坦的一道数学题: 有一条长长的阶梯, 若每步跨 2 阶, 则最后剩 1 阶; 若每步跨 3 阶, 则最后剩 2 阶; 若每步跨 5 阶, 则最后剩 4 阶; 若每步跨 6 阶, 则最后剩 5 阶; 若每步跨 7 阶则刚好 1 阶不剩。编程求这阶梯共有多少阶。

15. 计算并输出不定方程  $x^2 + y^2 = 10000$  ( $x \leq y$ ) 的全部自然数解。

## 实验五 常用算法

### 一、实验目的

1. 掌握数值积分的近似算法(矩形法、梯形法)。
2. 掌握求解一元方程根的多种算法(迭代法、二分法)。
3. 学习根据整数的一些性质求解数字问题。

### 二、实验内容

1. 以下程序用“辗转相除法”来求两个自然数的最大公约数。完善下列程序。

```
IMPLICIT none
INTEGER x,y,a,b,c,t
READ *,x,y
a=x
b=y
IF(a<b) THEN
    t=a
    a=b
    b=t
END IF
c=MOD(a,b)
DO WHILE( (1) )
    a= (2)
    b= (3)
    c= (4)
END DO
PRINT *,x,"和",y,"的最大公约数为:",b
END
```

思考:若求两个自然数的最小公倍数,程序如何修改?

2. 下面程序的功能是输出 2 到 500 之间的所有亲密数对。亲密数对的定义为:如果 M 的因子(包括 1 但不包括自身)之和为 N,且 N 的因子之和为 M,则称自然数 M 与 N 为一对亲密数,例如,6 的所有因子之和为  $1+2+3=6$ ,因此 6 与它自身构成一对亲密数。请在程序中的下划线处填入合适的内容。

```
DO M= (1)
    N=0
```

```

      (2)
      DO K=1, L
        IF (MOD(M,K).EQ.0) N=N+K
      END DO
      J=0
      (3)
      DO K=1, L
        IF (MOD(N,K).EQ.0) J=J+K
      END DO
      IF ((M.EQ.J).AND.(M.LE.N)) WRITE(*,*) M,N
      END DO
      END

```

3. 已知 
$$\begin{cases} F_1=0 \\ F_2=1 \\ F_n=F_{n-1}+2F_{n-2} \quad (n \geq 3) \end{cases}$$
 求:

(1) 10000 以内最大的一项;

(2) 输出  $F_6 \sim F_{10}$ 。

IMPLICIT NONE

INTEGER N,F0,F1,F

F0=0

F1=1

F=F0+2 \* F1

N=3

DO WHILE( (1) )

F0= (2)

F1= (3)

F=F0+2 \* F1

N=N+1

IF(N>=6, AND. N<=10) PRINT \*, F

END DO

PRINT \*, "10000 以内最大的一项为:", (4)

END

4. 利用迭代公式:

$$y_{n+1} = \frac{2}{3}y_n + \frac{x}{3y_n^2}$$

输入 x, 求  $y = \sqrt[3]{x}$  的近似值。初始值  $y_0 = x$ , 精确到  $10^{-6}$ 。

5. 矩形法和梯形法分别求  $\int_a^b (1+e^x)dx$ 。

当积分区间为 $[1.0, 2.0]$ 时,根据不同的区间数求定积分的值并填入下表。要求精确到小数点后第6位。

区 间 数 \ 方 法	矩形法数值积分值	梯形法数值积分值
100		
1000		
10000		

6. 用牛顿迭代法求方程  $f(x) = 2x^3 - 4x^2 + 3x - 7 = 0$  在  $x = 2.5$  附近的实根,直到满足  $|x_n - x_{n-1}| \leq 10^{-6}$  为止。

7. 求  $x = -2, x = 2, y = 0$  及  $y = 5 - x^2$  所包围图形的面积。

8. 输入一个十进制数,输出其二进制数。

9. 编程求  $s = x - \frac{1}{2} \cdot \frac{x^3}{3} + \frac{1}{2} \cdot \frac{3}{4} \cdot \frac{x^5}{5} - \frac{1}{2} \cdot \frac{3}{4} \cdot \frac{5}{6} \cdot \frac{x^7}{7} + \dots$

直到最后一项的绝对值小于  $10^{-15}$  为止。

10. 试用二分法求方程  $\sin(2x) + x - 1.9 = 0$  的在区间  $[1, 2]$  的一个实根  $X$ 。

11. 当自然数  $N$  表示为  $d_1 d_2 \dots d_k (k > 1)$  时,如总有  $d_i \geq d_{i-1} (i = 1, 2, 3, \dots, k-1)$  则称  $N$  为升序数。例如 13 和 179 是升序数,而 73 则不是。求 100~2000 之间为升序数的个数。

12. 已知  $a, b, c$  为自然数,且  $a > b > c, a + b + c < 100$ ,求满足  $\frac{1}{a^2} + \frac{1}{b^2} = \frac{1}{c^2}$  的倒勾股数共有多少组。



## 实验六 FORTRAN 数据类型

### 一、实验目的

1. 掌握双精度型、复型、字符型数据的描述方法及其运算。
2. 掌握四种数值型数据之间的转换和运算规则。

### 二、预备知识

1. 修改下列程序段中的错误：

(1) CHARACTER char \* 10

char=FORTRAN

(2) COMPLEX x

x=(3,4)

(3) LOGICAL luo

luo=true

(4) DATA x,y/1.,3.2,4.5/

(5) COMPLEX c

REAL a,b

c=(a\*2,3+b)

(6) REAL(4) x,y

REAL(2) t

t=x,x=y,y=t

2. 用 FORTRAN 表达式表示以下各题：

(1) 将 ch 中的第一个字母转换成小写字母；

(2) 将 D 中的一位十进制数字转换成对应的数字字符。

3. 执行下列程序：

```
CHARACTER*6 S1, S2
```

```
READ(*,*) S1
```

```
DO I=1, 6
```

```
IF(S1(I: I).GE.'A'.AND.S1(I: I).LE.'Z') THEN
```

```
    S2(I: I)=CHAR(ICHAR(S1(I: I))+32)
```

```
ELSE IF(S1(I: I).GE.'a'.AND.S1(I: I).LE.'z') THEN
```

```
    S2(I: I)=CHAR(ICHAR(S1(I: I))-32)
```

```
END IF
```

```
END DO
```

```
WRITE(*,*)S2
```

```
END
```

运行时,如果从键盘上输入:'ABCdef'↵,则输出为:

A) ABCdef      B) abcDEF      C) abc      D) DEF

4. 阅读下列程序:

```
COMPLEX A,B
```

```
REAL L
```

```
READ(*,*)A,B
```

```
L=ABS(A-B)
```

```
WRITE(*,*)'LENGTH=',L
```

```
END
```

该程序的功能是:\_\_\_\_\_

当输入为(1.5,1.0),(4.5,5.0)↵时,输出结果是:\_\_\_\_\_。

5. 阅读下列程序。

```
CHARACTER A,B*30
```

```
A='*'
```

```
DO I=1,5
```

```
  B=' '
```

```
  L=9-I
```

```
  DO J=1,2*I-1
```

```
    N=L+J
```

```
    B(N,N)=A
```

```
  END DO
```

```
  WRITE(*,*)B
```

```
END DO
```

```
END
```

该程序的执行结果是:\_\_\_\_\_。

### 三、实验内容

1. 完善程序:输入一个字符,若为小写字母,则输出其大写字母;若为大写字母,则输出其小写字母;若为数字,则原样输出该数字字符;否则,输出\*。

```
CHARACTER ch,chnew
```

```
READ*,ch
```

```
SELECT CASE(ch)
```

```
  CASE ( (1) )
```

```
    chnew=CHAR(ICHAR(ch)+32)
```

```
  CASE ( (2) )
```

```
    chnew=CHAR(ICHAR(ch)-32)
```

```
CASE ( (3) )  
    chnew=ch  
CASE DEFAULT  
    chnew='*'  
END SELECT  
PRINT *,chnew  
END
```

2. 用双精度数据计算:

$$\cos x \approx 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots + (-1)^n \frac{x^{2n}}{(2n)!}$$

直到第  $n$  项的绝对值小于  $10^{-15}$  为止。

3. 有一个三角形, 顶点为  $X, Y, Z$ , 其坐标分别为  $(1.5, 2.0), (4.5, 4.5), (18.0, 10.5)$ 。求三角形面积和三角形重心。

4. 编写程序统计输入的字符串中有多少个字符(不包含空格)。

## 实验七 格式输入输出

### 一、实验目的

1. 掌握各种格式编辑符的作用。
2. 熟悉格式语句与格式输入输出语句的相互作用。

### 二、预备知识

1. 写出下列编辑符的输出结果（其中□表示空格）。

变量值	-123	16888	45.67	-0.00987	-0.00987	56.345
编辑符	I3	I7	F7.3	F6.2	E10.2	E18.6E3
输出结果						
变量值	(2.3, -4.5)		, TRUE.	'Shift'	'PRINT'	'□Baby'
编辑符	2F6.2		L3	A5	A	A8
输出结果						

2. 写出下面格式输入后变量的值。

输入的数据	□-123	□1□68	45678	0.0887	□-00987	563E+2
编辑符	I3	I5	F7.3	F6.2	E7.2	E10.3
变量值						

3. 已知  $M=2000$ , 为了在屏幕上输出:

□□ \* \* \* 2000 \* \* \*

则应使用的格式输出语句为:

- A) WRITE(\*,10)M  
10 FORMAT(2X,' \* \* \* ',14,' \* \* \* ')
- B) WRITE(\*,10)  
10 FORMAT(3X,' \* \* \* M \* \* \* ')
- C) WRITE(\*,10)  
10 FORMAT(3X,' \* \* \* 2000 \* \* \* ')
- D) WRITE(\*,10)M  
10 FORMAT(1X,'2X',17,' \* \* \* ')

4. 阅读下列程序:

```
READ(*,100) M,N
K=M+N
WRITE(*,100) K
100 FORMAT(1X,2I3)
```

END

运行上述程序时,如果从键盘输入 123456789  $\swarrow$ ,则最后输出的结果为:

- A) 579                      B) 79                      C) 801                      D) 01

5. 阅读下列程序:

```
READ(*,100) X,Y
```

```
Z=X+Y
```

```
WRITE(*,100) Z
```

```
100 FORMAT(1X,2F4.1)
```

END

运行上述程序时,如果从键盘输入 543219876  $\swarrow$ ,则最后输出的结果为:

- A) \* \* \* \*                      B) 1419.7                      C) 141.9                      D) 19.7

6. 阅读下列程序:

```
READ(*,200) K,J
```

```
Y=MOD(K,J)/J
```

```
WRITE(*,*) Y
```

```
200 FORMAT(1X,I2,2X,I2)
```

END

运行上述程序时,如果从键盘输入 1248048965  $\swarrow$ ,则最后输出的 Y 值为:

- A) 0.3                      B) 0.25                      C) 0.5                      D) 0.0

7. 分析下面程序的输出结果。

```
IMPLICIT NONE
```

```
INTEGER a
```

```
REAL b,c
```

```
a=83
```

```
b=450.98
```

```
c=12.5
```

```
WRITE(*,10) a,b,c
```

```
WRITE(*,20) a,b,c
```

```
WRITE(*,30) a,b,c
```

```
10 FORMAT(1X,'a=',I3/,1X,'b=',F7.2/,1X,'c=',F7.2)
```

```
20 FORMAT(1X,T6,I5.3,T16,F7.2,T26,F7.2)
```

```
30 FORMAT(1X,'a=',I3:,'b=',F7.2:,'c=',F7.2)
```

END

### 三、实验内容

1. 以下程序在屏幕上输出一个矩形的九九乘法表。

```
DO i=1,9
```

```
    PRINT 100,(i*j,j=1,9)
```

```
END DO
```

```
100 FORMAT(9I5)
```

```
END
```

请对程序进行修改,使之在屏幕上输出一个直角三角形的九九乘法表。

2. 求  $\frac{1}{1 \times 2 \times 3} + \frac{1}{2 \times 3 \times 4} + \cdots + \frac{1}{100 \times 101 \times 102}$ , 要求输出结果保留两位小数。

3. 编写程序,输出由“\*”构成的等腰三角形,要求从键盘输入要打印的图案的行数和第一行“\*”位于第几列。如:输入打印的行数为4,第一行“\*”位于第15列,则屏幕上输出的图案如下:

第15列 → |

```
      *
    * * *
  * * * * *
* * * * * *
```

## 实验八 数 组

### 一、实验目的

1. 掌握数组的定义。
2. 掌握数组的赋值和输入输出方法。
3. 熟悉数组在内存中的存储顺序(按列存储)。
4. 掌握给数组赋初值的方法。
5. 掌握与数组有关的算法。

### 二、预备知识

1. 下列有关 FORTRAN 数组的说法正确的是:  
A) 数组定义语句只能定义一维或二维数组  
B) 数组元素下标不能出现负值  
C) 在赋值语句中出现的 A(3)肯定是数组元素  
D) 程序中使用的数组必须进行说明  
E) 在同一程序单位中,数组名可以与变量名相同

2. 阅读下列程序:

```
DIMENSION M(3,4)
DATA M/4*2,4*1,4*-1/
N=0
DO K=1,4
    N=N+M(1,K)
END DO
WRITE(*,*) N
END
```

运行上述程序后,输出的 N 的值为:

- A) 8                      B) 5                      C) 4                      D) 3

3. 阅读下列程序:

```
DIMENSION M(3,2)
READ(*,100)((K,L,M(I,J),J=1,2),I=1,3)
WRITE(*,200)((M(I,J),I=1,3),J=1,2)
100 FORMAT(20I1)
200 FORMAT(1X,2I1)
END
```

运行上述程序时,如果从键盘输入 123456789012345678 ↵,则最后输出的结果为:

A) 34	B) 39	C) 35	D) 34
78	56	74	56
12	28	68	78

4. 阅读下列程序:

```

IMPLICIT NONE
INTEGER, ALLOCATABLE :: s1(:, :, )
INTEGER m, n, i, j
READ *, m, n
ALLOCATE (s1(m, n))
IF (MOD(SIZE(s1), 2).EQ.0) THEN
    s1(1:m, 2, 1:n) = 1
    s1(2:m, 2, 1:n) = 0
ELSE
    s1(1:m, 1, n; 2) = 1
    s1(1:m, 2, n; 2) = 0
END IF
DO i = 1, m
    PRINT *, (s1(i, j), j = 1, n)
END DO
DEALLOCATE(s1)
END

```

运行上述程序时,如果从键盘输入:3,4 ↵,屏幕上输出:\_\_\_\_\_。

如果从键盘输入:3,3 ↵,屏幕上输出:\_\_\_\_\_。

### 三、实验内容

1. 利用冒泡排序法把输入的一系列无序的数据按由小到大的顺序排列。完善下列程序:

```

INTEGER A(10), TEMP
N = 10
READ *, A
PRINT *, A
DO K = 1, N - 1
    DO J = 1, (1)
        IF( (2) ) THEN
            TEMP = A(J)
            A(J) = A(J + 1)
            A(J + 1) = TEMP
        END IF
    END DO
END DO

```



```

    END DO
  END DO
  PRINT *, A
  END

```

2. 在 a 数组和 b 数组中各有 10 个由小到大的有序数, 以下程序把两个数组中的数按由小到大的顺序归并到 c 数组中, 并按每行输出 5 个数据的格式输出该数组中的有效元素。完善下列程序。

```

INTEGER A(10), B(10), C(20)
DATA A/1,2,5,7,12,24,35,36,40,80/
DATA B/3,4,6,8,10,12,25,28,36,56/
I=1
J=1
K=1
DO (1)
  IF(A(I) <= B(J). OR. J >= 11) THEN
    C(K)=A(I)
    IF(A(I) == B(J)) (2)
    I=I+1
  ELSE IF(A(I) > B(J). OR. I >= 11) THEN
    C(K)=B(J)
    J=J+1
  END IF
  K= (3)
END DO
WRITE(*,100) (4)
100 FORMAT(1X, 5I4)
END

```

3. 下列程序的功能是: 从键盘依次输入 10 个实数赋给一个长度为 10 的一维实型数组。在输入数据过程中, 要求保持该数组中的元素是从小到大排列的, 并且去掉值相同的数据, 最后在数组最后的剩余元素中均赋以 0。请在程序中的下划线处填入合适的内容。

```

DIMENSION A(10)
K=1
DO KK=1,10
  READ(*,*) X
  J=K-1
  DO WHILE(J.GT.0)
    IF (X.LT. A(J)) THEN
      (1)

```

```

      J=J-1
      ELSE
        EXIT
      END IF
    END DO
    IF ((J.GT.0).AND.(X.EQ.A(J))) THEN
      DO L=J+1,K-1
        A(L)=A(L+1)
      END DO
    ELSE
      (2)
      K=K+1
    END IF
  END DO
  DO KK=K,10
    (3)
  END DO
  WRITE(*,300) A
  300 FORMAT(1X,5F10.5)
  END

```

4. 下列 FORTRAN 程序的功能是求  $N!$ 。 $N!$  的值最大可以有 100 位。把  $N!$  的每一位数依次放在数组  $M$  之中并输出。请在程序中的下划线处填入合适的内容。

```

PROGRAM MAIN
  INTEGER M(100),L,KW
  DATA M/99*0,1/
  WRITE(*,*)'N=?'
  READ(*,*)N
  DO I=1,N
    L=0
    DO K= (1)
      KW=L+M(K)*I
      M(K)=MOD( (2) )
      L= (3)
    END DO
    IF(L.GT.0) THEN
      WRITE(*,*)N,'TOO BIG'
      STOP
    END IF
  END DO

```

```

END DO
K=1
DO WHILE(M(K).EQ.0)
    K=K+1
END DO
WRITE(*,*)N,(M(I),I= (4) )
END

```

5. 编写程序,其功能是:计算给定的 20 个数据与平均值之差的绝对值之和。

DIMENSION A(20)

DATA A/3.5,1.5,-2.5,4.5,2.5,3.0,5.5,-6.5,3.5,2.5, &  
1.5,2.5,4.5,-3.0,2.6,3.4,6.3,-3.8,5.2,-1.6/

6. 现有两个集合  $S=\{-4,8,5,2,3,7,9\}$ ,  $P=\{-4,9,8,2\}$ ,试编写程序判断 P 是否为 S 的子集。

7. 求数列 1,3,3,3,5,5,5,5,5,7,7,7,7,7,7,7,7,... 的第 40 项。

8. 设  $f(n)=1 \cdot 2 \cdot 3+4 \cdot 5 \cdot 6+\cdots+(3n-2) \cdot (3n-1) \cdot 3n$ ,求:

(1)  $f(20)$  (2)  $f(100)$

9. 有这样一个数列,该数列的前两项为输入的自然数,以后各项由其前面的两项之和按下列规则生成:若和小于 100,则它是下一项;若和大于 100,且最后两位数字不全为 0,则和除以 100 的商和余数依次作为下两项;否则以和除以 100 的商作为下一项。例如,若前两项为 15、17,则后面各项依次为 32,49,81,1,30,...,前 6 项之和为 195。

(1) 求该数列的第 10 项;

(2) 求前 15 项之和。

10. 计算 A,B 两个矩阵的积 C。最后计算并输出 C 矩阵主对角线元素的和 S。设:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

11. 设 X,Y 是 N 个元素的列向量,A,B 是  $N \times N$  方阵,编写程序计算  $Y=A * B * X$ 。

12. 找出一个  $m \times n$  的二维数组中的鞍点。所谓“鞍点”是指它在本行中值最大、在本列中值最小。在一个数组中可能存在多个鞍点,也可能不存在鞍点。输出鞍点的行号、列号,如无鞍点则输出“无”。

13. 分别按以下格式打印出杨辉三角形的前 n 行。

1	1
1    1	1    1
1    2    1	1    2    1
1    3    3    1	1    3    3    1
1    4    6    4    1	1    4    6    4    1
1    5    10    10    5    1	1    5    10    10    5    1

## 实验九 子程序

### 一、实验目的

1. 掌握语句函数的定义及引用方法、函数子程序(FUNCTION 子程序)和子例行程序(SUBROUTINE 子程序)的结构及调用方法。
2. 掌握 FORTRAN 程序单元间的数据传递方法。
3. 学习使用公用语句(COMMON 语句)、无名公用区和有名公用区及数据块子程序。

### 二、预备知识

1. 语句函数定义语句在程序内的位置是:

- A) 在程序块开头语句之后,END 语句之前
- B) 在说明语句之后,可执行语句之前
- C) 在说明语句之后,END 语句之前
- D) 在程序块开头语句之后,可执行语句之前

2. COMMON 语句的功能是:

- A) 给同一程序模块中的若干变量分配同一存储单元
- B) 给不同程序模块中的变量分配相同的存储单元
- C) 只能给不同程序模块中的相同名字的变量分配相同的存储单元
- D) 给同一程序模块中的相同类型变量分配相同存储单元

3. 下列语句函数的定义中正确的是:

- A)  $F(X,Y) = (X+Y)/(X*Y)+7.0$
- B)  $FUNCTION(I,J,K) = 3*I+2*J+0.5*K$
- C)  $H(A,B,C(I)) = SIN(A) + SIN(B) + C(I)$
- D)  $S(A,B,C) = A*B + S(A*A,B,C)$

4. 有如下子例行子程序:

```
SUBROUTINE SUB(K, A)
```

```
B=K+2
```

```
A=A+B
```

```
END
```

下列调用语句中正确的是:

- A) CALL SUB(N, N)
- B) CALL SUB(X, X)
- C) CALL SUB(N+2, X)
- D) CALL SUB(N, X+3)

5. 指出下列程序段中的错误。

```
(1) FUNCTION Y(A,B)
```

```
C=A+B
END
(2) EXTERNAL MOD
    CALL SUB(MOD,A,B)
(3) A=5.0
    Y=FUN(A)
    PRINT Y
    END
    FUNCTION FUN(I)
    FUN=I*1
    END
(4) INTEGER I,J,SUB
    I=2
    J=8
    CALL SUB(I,J)
(5) INTEGER::A(3)=3
    Y=FUN(A)
    PRINT Y
    END
    FUNCTION FUN(X)
    INTEGER X
    FUN=X+1
    END
(6) INTEGER::A(2,3)=1
    CALL SUB(A(2,3))
    END
    SUBROUTINE SUB(X)
    INTEGER X(6)
    PRINT *,X
    END
```

6. 阅读下列程序:

```
EXTERNAL P
CALL S(P,3.0,4.0,T)
WRITE(*,100) T
100 FORMAT(1X,F5.1)
END
FUNCTION P(X,Y)
P=X*X+Y*Y
```

```

END
SUBROUTINE S(F,A ,B ,C)
C=F(A,B )
C=F(B,C)
END

```

运行上述程序后,输出的 T 的值为:\_\_\_\_\_。

7. 阅读下列程序:

```

INTEGER B(4,5),S
DATA B/11,20,15,1,9,12,18,2,6,8,&
      10,3,22,15,21,3,17,9,35,4/
K=S(B,4,5)
WRITE(*,*) K
END
INTEGER FUNCTION S(A,M,N)
INTEGER A(M,N),D,P
DO L=1,M
  P=A(L,1)
  K=1
  DO J=1,N
    IF (A(L,J).LT.P) THEN
      P=A(L,J)
      K=J
    END IF
  END DO
  D=A(1,K)
  DO J=1,M
    IF (A(J,K).GT.D) D=A(J,K)
  END DO
  IF (P.EQ.D) S=P
END DO
END

```

运行上述程序后,输出的 K 的值为:

A) 1                      B) 10                      C) 35                      D) 3

8. 下列程序段在无名公用区中实型元素的个数是:

```

DIMENSION A(2), B(3), C(2,3)
COMMON A, X
COMMON B

```

A) 6                      B) 5                      C) 11                      D) 3

9. 阅读下列程序:

```
COMMON K,M,N,L
K=4
M=5
CALL AAA
WRITE(*,100) K,M,N
100 FORMAT(1X,I2)
END
SUBROUTINE AAA
COMMON J,K,M,N
J=J+K
M=J*K
N=J+M
END
```

运行上述程序后,输出的结果为:

A) 4	B) 4	C) 9	D) 9
5	5	5	5
29	20	45	29

10. 阅读下列程序:

```
DIMENSION A(4,3)
COMMON /P/A
P=0.0
DO K=1,3
  DO J=1,3
    IF (P.LE.5.0) P=P+A(J,K)
  END DO
END DO
WRITE(*,*) P
END
BLOCK DATA ABC
DIMENSION B(4,3)
COMMON /P/B
DATA B/4*4.0,4*-3.0,4*2.0/
END
```

运行上述程序后,输出的 P 值为:

A) 8.0	B) 7.0	C) 5.0	D) 4.0
--------	--------	--------	--------

11. 阅读下列程序:

```
CHARACTER STR(2,3)*2
```

```

STR=" "
CALL SUB(STR)
PRINT *,STR(2,2)
END
SUBROUTINE SUB(A)
CHARACTER A(3,5)*1
DO N=1,3
    DO M=1,5
        A(N,M)=CHAR(ICHAR("a")+N+M)
    END DO
END DO
END

```

运行上述程序后,输出的结果为:\_\_\_\_\_。

### 三、实验内容

1. 下列子程序的功能是:从键盘输入一个大于6的偶数,然后将该偶数表示成两个素数之和。请在程序中的下划线处填入合适的内容。

```

WRITE(*,*)"请输入一个偶数:"
READ(*,*)K
J=3
DO WHILE( _____ .AND. J. LE. K/2)
    J=J+2
END DO
WRITE(*,200) K, J, K-J
200 FORMAT(1X, I4, '=', I4, '+', I4)
END
FUNCTION KF(K)
M=SQRT(1.0*K)
KF=1
_____(2)_____
DO WHILE(MOD(K, J). NE. 0. AND. J. LE. M)
    J=J+1
END DO
_____(3)_____
END

```

2. 用梯形公式求数值积分的近似公式为  $\int_a^b f(x) dx = \frac{b-a}{2} [f(a) + f(b)]$ , 取  $f(x) = s_0 + s_1 x + s_2 x^2$ , 对于不同的系数  $s_0, s_1, s_2$ , 和积分上下限  $a, b$ , 计算近似积分值并输出结果。



其中  $f(x)$  用函数子程序实现, 数值积分用子程序实现, 写成下列程序。请在程序中的下划线处填入合适的内容。

```

    (1)
COMMON S0,S1,S2
WRITE(*,*)'S0,S1,S2='
READ(*,*)S0,S1,S2
READ(*,*)A,B
CALL SUB( (2) )
WRITE(*,*)'S0=',S0,'S1=',S1,'S2=',S2
WRITE(*,*)'A=',A,'B=',B
WRITE(*,*)'FAB=',FAB
END
FUNCTION F(X)

```

```

    (3)
F=S0+S1*X+S2*X*X
END
SUBROUTINE SUB(A,B,F,FAB)
COMMON S0,S1,S2
FAB=0.5*(B-A)*(F(A)+F(B))
END

```

3. 若 Fibonacci 数列的第  $n$  项记为  $\text{fib}(a,b,n)$ , 则有下面的递归定义:

```

fib(a,b,1) = a
fib(a,b,2) = b
fib(a,b,n) = fib(b,a+b,n-1)    (n>2)

```

用递归方法求 5000 之内最大的一项。

```

IMPLICIT NONE
INTEGER n,a,b,fib
READ *,a,b
n=1
DO WHILE( fib(a,b,n)<5000 )
    n=n+1
END DO
PRINT 10,'5000 之内的最大项是第', (1) , '项, 其值为:', (2)
10 FORMAT(1X,A20,I2,A12,I4)
END
    (3)
INTEGER a,b,n,fib
SELECT CASE(n)

```

CASE(1)

fib=a

CASE(2)

fib=b

CASE DEFAULT

fib= (4)

END SELECT

END

4. 求函数  $f(x) = \frac{x^2}{\sqrt{1+2x+x^2}}$  在  $x=1, 2, 3, 4, 5$  处的值。要求用语句函数实现。

5. 设  $\text{sh}(x) = \frac{e^x - e^{-x}}{2}$ , 求  $\text{sh}(2) - \text{sh}(3)$ 。

6. 已知  $y = \frac{f(f(a)+1)}{f(2a)}$ , 其中  $f(x) = \begin{cases} \frac{1}{2x} & x \leq 1 \\ x^2 - 5x & x > 1 \end{cases}$

输入 a, 求 y 的值。

7. 编写求阶乘及累加和的函数, 调用函数求:

$$S = \sum_{n=1}^m \left( \prod_{i=1}^n i + \sum_{i=1}^n i \right)$$

当  $m=10$  时,  $S =$  \_\_\_\_\_。

8. 一个自然数是素数, 且它的数字位置经过任意对换后仍为素数, 则称这样的数为绝对素数, 例如 13。编写一个判断素数的子程序, 调用该子程序找出 10~99 之间所有的绝对素数。

9. 设函数  $m(x)$  定义为  $m(x) = \begin{cases} x-10 & x > 100 \\ e^{x/10} & x \leq 100 \end{cases}$ , 求:

(1)  $m(m(102))$

(2)  $\sum_{n=21}^{10} m(n)$

10. 编写函数子程序 GCD 求两个数的最大公约数, 求最大公约数的算法如下: 把两个数中大的那个数作为被除数, 小的那个数作为除数, 两数相除得一余数, 把余数去除以除数得一新的余数。不断重复这一过程直到余数为零, 这时的除数就是两个数的最大公约数。

(1) 调用此函数求 1260, 198, 72 三个数的最大公约数;

(2) 调用此函数求 1896, 960, 256, 96 四个数的最大公约数。

11. 已知  $y = \frac{f(a)+f(b)}{a+b}$ , 其中:

$$f(x) = x + \frac{x^2}{1+2} + \frac{x^3}{1+2+3} + \cdots + \frac{x^N}{1+2+3+\cdots+n}$$

直到第 n 项的值小于  $10^{-6}$  为止, 要求  $|x| < 1$ 。

当  $a=0.8, b=0.9$  时,  $y =$  \_\_\_\_\_。

12. 编写牛顿迭代法求方程的根的函数,调用此函数求方程  $x^3 - 3x + 1 = 0$  在  $(0, 1)$  内的解。请在其中下划线处填入适当的内容。

```
EXTERNAL FUN1
READ *, X0
X1 = X0
(1)
PRINT *, X1
END
SUBROUTINE SUB(F, X)
DO WHILE( (2) )
    X0 = X
    X = (3)
END DO
END
FUNCTION FUN1(X)
F = X * * 3 - 3 * X + 1
F1 = 3 * X * X - 3
FUN1 = F/F1
END
```

13. 函数子程序 DTOH 能够将一个十进制正整数转换为一个表示十六进制数的字符串,其长度不超过 8。例如 127 转换为 7F。算法是:将十进制整数除以 16 取余数,将所得的商再除以 16 取余数,如此反复,直到商等于零为止,最后按照逆序输出所得的各个余数。请在其中下划线处填入适当的内容。

```
CHARACTER * 8 FUNCTION DTOH(DEC)
INTEGER DEC, R, POS
POS = 9
DTOH = ''
DO WHILE( (1) )
    (2)
    IF(POS. GE. 1) THEN
        R = MOD(DEC, 16)
        IF(R. LT. 10) THEN
            (3) = CHAR(ICHAR('0') + R)
        ELSE
            DTOH(POS, POS) = CHAR(ICHAR('A') + R - 10)
        END IF
    ELSE
        DTOH = ' * * * * * '
```

```
EXIT
ENDIF
DEC=DEC/16
END DO
END
PROGRAM DECHEX
CHARACTER * 8 DTOH
WRITE(*, '(A)') DTOH(127)
END
```

14. 编写求数组元素之和的函数。已知 A,B 均为数组,调用函数求 A 数组元素之和,B 数组元素之和。要求通过 COMMON 语句把 A,B 数组的值传给函数,且通过数据块子程序给 A,B 数组赋值。

## 实验十 结构体

### 一、实验目的

1. 掌握结构体的概念、定义和格式。
2. 掌握结构体的引用和初始化方法。
3. 掌握结构体数组的定义。

### 二、预备知识

1. 阅读下列程序。

```
IMPLICIT NONE
TYPE teacher
    INTEGER tno
    CHARACTER tname * 8
    INTEGER birth
    LOGICAL sex
END TYPE
TYPE(teacher); teach1=teacher(1001,"张云",1963,.false.)
IF(teach1.sex) THEN
    PRINT *,2002-teach1.birth
ELSE
    PRINT *, " * * * "
END IF
END
```

运行程序,其输出结果为: \_\_\_\_\_。

2. 阅读下列程序。

```
IMPLICIT NONE
TYPE person
    CHARACTER name * 8
    REAL salary
    REAL award
    REAL cost
END TYPE
INTEGER i,j,NUM
PARAMETER (NUM=10)
```

```
TYPE(person), DIMENSION(NUM); clerk
DO i=1, NUM
    READ *, clerk(i), name, clerk(i), salary, clerk(i), award, clerk(i), cost
END DO
DO j=1, NUM
    IF(clerk(j), salary+clerk(j), award-clerk(j), cost >= 1200) THEN
        PRINT *, clerk(j), name, clerk(j), salary, clerk(j), award, clerk(j), cost
    END IF
END DO
END
```

运行程序,输入以下数据:

"Lister",876,451,123

"Angel",746,523,23

"Billy",912,551,243

"Henry",926,660,314

"Ervin",676,481,103

"Ford",816,351,89

"George",936,451,145

"Mailer",766,480,98

"Philip",880,560,168

"Victor",660,440,86

输出的结果为: \_。

### 三、实验内容

1. 试定义一个派生类型数据,要求包括货号、品名、单价、数量。
2. 使用第1题中已经定义的派生类型,对给出详细资料的三十种的货物资料进行查找,列出采购金额超过10000元的货物。采购金额 = 单价 × 数量。

## 实验十一 指 针

### 一、实验目的

1. 掌握指针的概念、定义格式和使用方法。
2. 理解动态变量的含义。
3. 了解链表的特点。

### 二、预备知识

1. 阅读下列程序。

```
IMPLICIT NONE
CHARACTER, POINTER::p1
CHARACTER, POINTER::p2
CHARACTER, TARGET::ch(2)
DATA ch/'A','B'/
p1=>ch(1)
p2=>ch(2)
p1=p2
PRINT 10,p1,p2
10 FORMAT(1X,2A2)
END
```

执行程序,在屏幕上的输出结果为:\_\_\_\_\_。

2. 修改下列程序段中的错误。

(1) IMPLICIT NONE

```
REAL, POINTER::p
INTEGER, TARGET::b
READ *, b
p=>b
```

(2) INTEGER, POINTER::p(:)

```
INTEGER::w(5)
READ *, w
p=>w
```

(3) INTEGER, POINTER::p(:, :)

```
INTEGER, TARGET::w(6)
READ *, w
```

p=>w

3. 阅读下列程序。

```
IMPLICIT NONE
INTEGER, POINTER :: s1(:, :)
INTEGER, TARGET :: a(2,3)
DATA a/1,2,3,4,5,6/
s1=>a
s1(1:2,1:3,2)=9
PRINT 10,a
10 FORMAT(1X,6I3)
END
```

执行程序,在屏幕上的输出结果为:\_\_\_\_\_。

4. 阅读下列程序。

```
IMPLICIT NONE
INTEGER, POINTER :: s1(:, :)
INTEGER, TARGET :: w(5,5)
INTEGER i,j,x(5)
DATA w/5*1,5*2,5*3,5*4,5*5/
DATA x/5*10/
s1=>w
DO i=1,5
s1(1,i,i,5)=x(i)+w(i,i)
END DO
PRINT 10,((w(i,j),j=1,5),i=1,5)
10 FORMAT(1X,5I3)
END
```

执行程序,在屏幕上的输出结果为:\_\_\_\_\_。

### 三、实验内容

1. 编写程序,让两个指针分别指向不同的目标变量。以不同的顺序,交换两个指针所指向的目标变量,输出交换后变量的值。

2. 以下程序用于在屏幕上输出如下图形:

```
1
2 1
3 2 1
4 3 2 1
5 4 3 2 1
IMPLICIT NONE
```



```

INTEGER, POINTER :: p(:)
INTEGER, TARGET :: x(5,5)
INTEGER i
x=0
DO i=1,5
    p=>x(1,i,1)
    p=p+1
PRINT *,p
END DO
END

```

请将程序略作修改,用于输出下列图形:

(1)	(2)
1 1 1 1 1	1
2 2 2 2	2 2
3 3 3	3 3 3
4 4	4 4 4 4
5	5 5 5 5 5

3. 建立一个链表,每一个结点包括的成员为:学号、平均成绩。要求链表包括 8 个结点,从键盘输入结点中的有效数据,然后把这些结点的数据打印出来。

4. 建立链表,开辟 8 个结点,每个结点包括一个整型数,现要求删除第 5 个结点。

## 实验十二 文 件

### 一、实验目的

1. 掌握文件与记录的概念。
2. 掌握文件的打开与关闭。
3. 掌握文件的存取方法。
4. 学习使用内部文件的方法。

### 二、预备知识

1. 执行下列语句：

```
OPEN(3, FILE='FILE.DAT')
```

之后,FILE.DAT 文件是:

- |            |            |
|------------|------------|
| A) 顺序有格式文件 | B) 直接有格式文件 |
| C) 顺序无格式文件 | D) 直接无格式文件 |

2. 下列关于直接文件操作说法中不正确的是:

- |                    |
|--------------------|
| A) 直接文件的所有记录的长度都相等 |
| B) 直接文件不能按记录的顺序读取  |
| C) 直接文件不能按表控格式存取   |
| D) 顺序写入的文件都不能直接读取  |

3. 阅读下列程序:

```
F(X,Y)=X * X + Y * Y
```

```
OPEN(8,FILE='FD.DAT',STATUS='NEW',ACCESS='DIRECT',&  
FORM='UNFORMATTED',RECL=16)
```

```
DO K=1,6
```

```
  P=F(REAL(K),REAL(K+1))
```

```
  WRITE(8,REC=K) P
```

```
END DO
```

```
CLOSE(8)
```

```
OPEN(6,FILE='FD.DAT',STATUS='OLD',ACCESS='DIRECT',&  
FORM='UNFORMATTED',RECL=16)
```

```
DO K=2,6,2
```

```
  READ(6,REC=K) T
```

```
  WRITE(*,100) T
```

```
END DO
```

```
100 FORMAT(1X,F4.1)
```

```
CLOSE(6)
```

```
END
```

运行上述程序后,输出的结果为:

A) 5.0	B) 5.0	C) 13.0	D) 41.0
13.0	25.0	41.0	61.0
25.0	61.0	85.0	85.0

4. 阅读下列程序。

```
CHARACTER STR * 10
```

```
OPEN(8,FILE='FILE1.DAT')
```

```
WRITE(8,100) (CHAR(I),I=ICHAR("A"),ICHAR("Z"))
```

```
CLOSE(8)
```

```
OPEN(6,FILE='FILE1.DAT',ACCESS='DIRECT',&
```

```
FORM='FORMATTED',RECL=10)
```

```
DO K=1,3
```

```
READ(6,200,REC=K)STR
```

```
WRITE(*,300) STR
```

```
END DO
```

```
100 FORMAT(10A1)
```

```
200 FORMAT(A5)
```

```
300 FORMAT(1X,A5)
```

```
CLOSE(6)
```

```
END
```

运行上述程序后,输出的结果为:\_\_\_\_\_。

5. 阅读下列程序。

```
PROGRAM INTERNALFILE
```

```
CHARACTER * 10 STR1, STR2(3)
```

```
INTEGER TRAN1
```

```
LOGICAL TRAN2
```

```
STR1='7890'
```

```
VALUE=123.45
```

```
TRAN2=.FALSE.
```

```
READ (STR1,'(I4)') TRAN1
```

```
WRITE(STR2(1),'(F7.3)') VALUE
```

```
WRITE(STR2(2),'(L7)') TRAN2
```

```
WRITE(STR2(3),'(A3,A7)') STR1, STR2(1)
```

```
WRITE(*,'(1X,I4,3(1X,A10))') TRAN1,STR2
```

```
END
```

运行上述程序后,输出的结果为:\_\_\_\_\_。

6. 阅读下列程序。

```
CHARACTER S*10
OPEN(8,FILE='FILE1.DAT')
WRITE(8,100) (I,I=1,50,2)
BACKSPACE(8)
BACKSPACE(8)
DO K=1,3
    READ(8,200,ADVANCE="no",EOR=300)S
    WRITE(*,"(1X,A)") S
END DO
100 FORMAT(10I2)
200 FORMAT(A10)
300 WRITE(*,"(1X,A)") "END OF RECORD"
CLOSE(8)
END
```

运行上述程序后,输出的结果为:\_\_\_\_\_。

7. 阅读下列程序。

```
INTEGER(4) S
OPEN(8,FILE='FILE1.DAT')
WRITE(8,100) (I,I=1,50,2)
REWIND(8)
K=1
DO WHILE(.NOT.EOF(8))
    READ(8,200)S
    WRITE(*,"(1X,I5)") S
    K=K+1
END DO
100 FORMAT(10I2)
200 FORMAT(I5)
CLOSE(8)
END
```

运行上述程序后,输出的结果为:\_\_\_\_\_。

### 三、实验内容

1. 从键盘输入若干字符串,并把这些字符串存放到文件 FILE1.TXT 中。
2. 以下程序用于产生 1000 个随机实数生成源数据文件 FILE21.DAT。

```
DIMENSION A(1000)
```

```

OPEN(1,FILE='FILE21.DAT',STATUS='UNKNOWN')
DO I=1,1000
    CALL RANDOM(Y)
    A(I)=Y
END DO
WRITE(1,100) A
100 FORMAT(1X,5F10.6)
CLOSE(1)
END

```

其中 RANDOM() 用于产生随机数。

请分别采用无格式顺序存取方式、有格式直接存取方式、无格式直接存取方式、有格式二进制存取方式、无格式二进制存取方式来修改上述程序,创建文件 FILE22.DAT~FILE26.DAT 写入数据。比较六个文件的长度。

3. 将有格式直接存取文件 FILE23.DAT 中的数据按从小到大的顺序重新写入 FILE23.DAT。

4. 从键盘输入一个记录号,删除有序文件 FILE23.DAT 中的指定记录。

5. 编写程序,其功能是:从键盘输入一个字符串(20 个字符以内),以其作为文件名,扩展名为 .TXT,查询磁盘上有无同名文件。如果有,则在原文件尾添加一句话“This is the end.”;如果没有,则创建一个有格式直接存取文件,存放某单位的客户信息,记录格式如下:

数据项名	姓名	性别	邮政编码	地址
类型	A10	L1	I6	A30

6. 假设上题的文件中存有 10 个客户的信息,编写程序查询所有男性客户的信息,并显示在屏幕上。

7. 下列程序的功能是:有序直接文件的插入。有序文件是按关键字项由小到大排列,第一个记录是该有序文件记录的个数,文件实际内容是从第二个记录开始。

请在程序中的下划线处填入合适的内容。

```

SUBROUTINE INONE(IX)
INTEGER IX,P,KA
OPEN(8,FILE='IN.DAT',ACCESS='DIRECT',&
    FORM='FORMATTED',RECL=5)
100 FORMAT(I5)
N=_(1)
READ(8,100,end=55) N
55 IF(N.EQ.1) WRITE(8,100,REC=1)N
DO P=N,2,-1
    READ(8,100,REC=P) KA

```

```

IF(KA.GT.IX) THEN
    WRITE(8, (2) )KA
ELSE
    EXIT
END IF
END DO
WRITE(8, (3) )IX
N=N+1
WRITE(8,100,REC=1)N
END

```

8. 对一个已经存在的有格式顺序存取文件,修改其中的第  $m$  个记录。

9. A,B,C 三个文件中的数据均为按序存放,一个数占一个记录。编写程序把三个文件中的内容合并成一个文件,合并后的文件中的数据仍然有序,相同的数只保留一个。

10. A,B 两个文件中都存放着整数,一个数占一个记录,每个文件中记录都不相同。把既在 A 文件中存在,又在 B 文件中存在的记录找出放在 C 文件中。

11. 读出当前文件夹下文件 README.TXT 中的内容,显示在屏幕上。请在程序中的下划线处填入合适的内容。

```

CHARACTER CH(10)*10
OPEN(5,FILE="README.TXT")
DO WHILE( (1) )
    CH=""
    DO N=1,10
        READ(5,'(A10)',EOR=100, (2) ) CH(N)
    END DO
    100 WRITE(*,200) (CH(K), (3) )
END DO
200 FORMAT(1X,10A10)
END

```

## 实验十三 FORTRAN 图形用户界面设计

### 一、实验目的

1. 学习如何创建 QuickWin 窗口。
2. 掌握图形的绘制方法。
3. 掌握使用资源编辑器设计对话框。
4. 学习开发 Windows 应用程序界面。

### 二、预备知识

1. 下列说法中哪些是正确的,哪些是错误的。

(1) 使用 QuickWin 特性的任何程序必须包含 USE MSFLIB 语句来包含图形库。

(2) USE MSFLIB 语句可以出现在其他说明语句之前,也可以位于其他说明语句之后。

(3) FL32 /MWs myapp. f90 用于创建一个名为 myapp 的 QuickWin 应用。

2. 阅读下列程序。

```
USE MSFLIB
INTEGER(2) x, y, wx, wy, status
TYPE (xycoord) xy
x=100
y=100
CALL MOVETO (x, y, xy)
status = SETCOLORRGB(#0000FF)
DO i = 10, 30, 10
  SELECT CASE (i)
    CASE(10)
      Wx = 50
      Wy = -50
    CASE(20)
      Wx = 50
      Wy = 50
    CASE (30)
      Wx = -100
      Wy = 0
  END SELECT
```

```

    x=x+wx
    y=y+wy
    status = LINE TO( x, y )
END DO
END

```

运行上述程序后,输出的结果为:\_\_\_\_\_。

### 三、实验内容

1. 设计一个窗口,绘制两个同心圆,外圆为红色空心圆,内圆为白色实心圆。
2. 设计一个窗口,绘制正弦曲线。请在程序中的下划线处填入合适的内容。

```

USE MSFLIB
INTEGER(2) x, y
INTEGER(4) color, status, w_height
TYPE (QWINFO) winfo, qwi
PARAMETER (PI=3.141593)
qwi.type=QWIN$MAX
status=SetWindowSize(QWIN$FRAMEWINDOW, qwi)
status = GETWSIZEQQ(QWIN$FRAMEWINDOW, QWIN$SIZEMAX, &
    winfo)
OPEN (10, FILE= (1) )
winfo.TYPE = QWIN$SET
status = SETWSIZEQQ(10, winfo)
w_height=winfo.h
color = #0000FF
DO i = 0, 360, 5
    x=i
    y=-SIN( (2) ) * 100 + w_height/2
    status = (3)
END DO
READ(10, *)
END

```

3. 输入下列程序,观察运行结果。

```

USE MSFLIB
INTEGER(2) status
INTEGER(4) color
TYPE (xycoord) xy
status = INITIALIZEFONTS ( )
status = SETFONT ('Arial" h18w10i')

```



```
color = SETCOLORRGB(0000FF)
CALL MOVETO (INT2(10), INT2(30), xy)
CALL OUTGTEXT('Fortran')
END
```

要使文字逆时针旋转  $90^\circ$ , 程序该如何修改。

4. 设计一个窗口, 删除菜单中 File 菜单的第三个菜单项, 在最后增加一个菜单项 Memo。

5. 编制程序程序, 显示一个信息框, 提示“继续”和“退出”。点击“继续”按钮, 则从磁盘上读入一个图形文件显示在屏幕上, 点击“退出”按钮, 则中断程序运行。

6. 设计一个学生信息录入系统, 录入的信息包括学号、姓名、性别、出生年月、家庭住址。要求录入界面中学号、姓名、家庭住址用编辑框录入, 性别用单选按钮录入, 出生年月用下拉列表框录入, 并把信息保存至一个文件中。

7. 设计一个学生信息查询系统, 可以从上题建立的文件中按年龄查询出 20 岁以下学生的详细信息, 并在屏幕上列出这些学生的信息。

## 实验十四 综合实验

### 一、实验目的

综合运用所学知识,掌握利用 FORTRAN 语言解决实际问题的方法。

### 二、实验内容

#### 1. 线性系统对任意外力响应的数值解法

如图 2.1 所示的单自由度线性系统。设该系统的弹簧系数  $k=10$  磅/英寸,阻尼系数  $c=0$ ,固有周期  $\tau=1.2$  秒。所加的外力  $Q$  随时间的变化如图 2.2 所示。其中  $Q$  的单位为磅, $t$  的单位为秒。已给定初始位移  $x_0=0$ ,初始速度  $\dot{x}_0=0$ 。试求该系统在  $0 \leq t \leq 1.6$  秒内的位移响应。

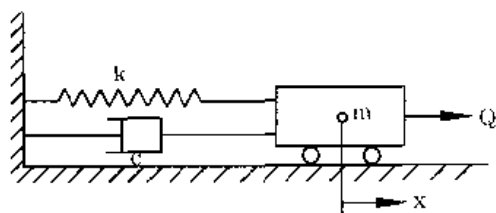


图 2.1 单自由度线性系统

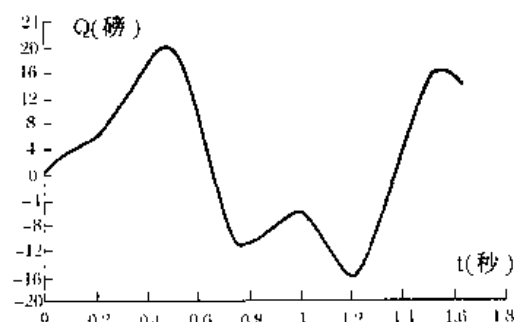


图 2.2 作用于某无阻尼线性系统的外力

#### (1) 把实际问题归纳为数学问题,构造数学模型

为了说明问题,先考虑一般的情况。在图 2.1 所示系统中,系统运动的微分方程为:

$$m \ddot{x} = -c \dot{x} - kx + Q \quad (2.1)$$

其中, $m$  为质量, $x$  为位移, $t$  为时间, $c$  为阻尼系数, $k$  为弹簧系数, $\dot{x}$  为速度, $\ddot{x}$  为加速度, $Q$  为外力,通常它是时间的函数。

用  $m$  去除方程(2.1)的两边,并引进下列符号:

$$p^2 = \frac{k}{m}, 2n = \frac{c}{m}, q = \frac{Q}{m} \text{ (单位质量上的外力)}$$

重新排列诸项,得到:

$$\ddot{x} + 2n \dot{x} + p^2 x = q \quad (2.2)$$

为了求解(2.2)的微分方程,先考虑  $q=0$  (无外力)的简单情况,此时

$$\ddot{x} + 2n \dot{x} + p^2 x = 0 \quad (2.3)$$

可以应用求解线性常系数微分方程的方法对方程(2.3)求解。假设此方程的解为下列形式:

$$x = Ce^{rt} \quad (2.4)$$

其中,  $e$  为自然对数的底,  $t$  为时间,  $C$  为待定常数,  $r$  为一常数。

$r$  的值必须由(2.4)满足方程(2.3)这一条件来确定, 将(2.4)代入方程(2.3), 得到:

$$r^2 + 2nr + p^2 = 0 \quad (2.5)$$

解此代数方程, 可得:

$$r = -n \pm \sqrt{n^2 - p^2} \quad (2.6)$$

当  $n^2 < p^2$  时, 得到两个共轭复根:

$$r_1 = -n + ip_d \text{ 和 } r_2 = -n - ip_d$$

其中,  $p_d = \sqrt{p^2 - n^2}$  称为有阻尼的振动系统的角频率, 其相应的周期为:

$$\tau_d = \frac{2\pi}{p_d} = \frac{2\pi}{p} \cdot \frac{1}{\sqrt{1 - (n^2/p^2)}}$$

将这两个根代入(2.4), 即可得方程(2.3)的两个解。而这两个解之和或之差乘以任一常数也将是方程(2.3)的一个解。按照这种方法, 求得:

$$x_1 = \frac{C_1}{2} (e^{r_1 t} + e^{r_2 t}) = C_1 e^{-nt} \cos p_d t$$

$$x_2 = \frac{C_2}{2i} (e^{r_1 t} - e^{r_2 t}) = C_2 e^{-nt} \sin p_d t$$

将这两个解加起来, 便得到方程(2.3)的下列形式的通解:

$$x = e^{-nt} (C_1 \cos p_d t + C_2 \sin p_d t) \quad (2.7)$$

上式中的待定常数  $C_1$  和  $C_2$  必须由初始条件来确定。为了确定方程(2.7)中的常数  $C_1$  和  $C_2$ , 设初始位移为  $x_0$ , 初始速度为  $\dot{x}_0$ , 代入方程(2.7)及其对时间的一阶导数中, 得到:

$$C_1 = x_0 \quad \text{和} \quad C_2 = \frac{\dot{x}_0 + nx_0}{p_d} \quad (2.8)$$

再将式(2.8)代入式(2.7)可得通解为:

$$x = e^{-nt} \left( x_0 \cos p_d t + \frac{\dot{x}_0 + nx_0}{p_d} \sin p_d t \right) \quad (2.9)$$

式中与  $\cos p_d t$  成比例的第一项仅取决于初始位移  $x_0$ , 与  $\sin p_d t$  成比例的第二项取决于初始位移  $x_0$  和初始速度  $\dot{x}_0$ 。

现在, 再回到方程(2.2), 讨论  $q$  为任意外力的情况。假设单位质量上的力  $q$  是虚时间变量  $t'$  的函数, 即:

$$q = \frac{Q}{m} = \frac{F(t')}{m} = f(t') \quad (2.10)$$

如图 2.3 所示。那么, 在任意瞬间  $t'$  处, 可以计算出图中阴影窄线条所代表的冲量增量  $q dt'$ 。不管还有什么别的力作用在它上面, 也不管在瞬时  $t'$  处它的位移和速度是多少, 此冲量给予每个单位质量上的速度增量等于  $d\dot{x} = q dt'$ 。如果将这个速度增量当作一个在瞬时  $t'$  处的初始速度那样来处理, 并利用方程(2.9), 就得出结论, 在以后的任意时刻  $t$  处, 这个系统的位移增量将是

$$dx = e^{-n(t-t')} \frac{q dt'}{p_d} \sin p_d (t - t') \quad (2.11)$$

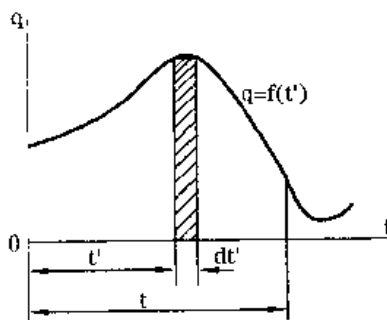


图 2.3  $q$  为任意外力时的情况

因为在  $t'=0$  和  $t'=t$  之间, 每一个冲量增量  $qdt'$  都具有这种效应, 于是, 由于这个干扰力  $q$  的连续作用结果, 得到总的位移为:

$$x = \frac{e^{-nt}}{p_d} \int_0^t e^{nt'} q \sin p_d(t-t') dt' \quad (2.12)$$

这种数学形式, 在数学上称为杜哈米(Duhamel's)积分。

式(2.12)代表了在时间间隔 0 至  $t$  期间作用在系统上的外力  $q$  所产生的整个位移, 此公式在研究振动系统对任意外力的响应时是非常有用的。如果要把  $t=0$  处的初始位移  $x_0$  和初始速度  $\dot{x}_0$  的影响考虑进去, 只需要将方程(2.12)与给出了初始条件的解(2.9)加起来即可, 这样, 系统的全解为:

$$x = e^{-nt} \left[ x_0 \cos p_d t + \frac{\dot{x}_0 + nx_0}{p_d} \sin p_d t + \frac{1}{p_d} \int_0^t e^{nt'} q \sin p_d(t-t') dt' \right] \quad (2.13)$$

如果略去阻尼, 取  $n=0$  和  $p_d=p$ , 对于这种情况, 方程(2.12)可简化为:

$$x = \frac{1}{p} \int_0^t q \sin p(t-t') dt' \quad (2.14)$$

而方程(2.13)可简化为:

$$x = x_0 \cos pt + \frac{\dot{x}_0}{p} \sin pt + \frac{1}{p} \int_0^t q \sin p(t-t') dt' \quad (2.15)$$

作为一个例子, 考虑下面一种特殊情况, 即所加的外力是一个大小为  $Q_1$ 、持续时间为  $t_1$  的矩形脉冲(图2.4),

那么, 在时间  $0 \leq t \leq t_1$  内,  $q = q_1 = \frac{Q_1}{m} = \text{常数}$ , 代入方程(2.12), 可得:

$$x = \frac{q_1 e^{-nt}}{p_d} \int_0^t e^{nt'} \sin p_d(t-t') dt' \quad (0 \leq t \leq t_1) \quad (2.16)$$

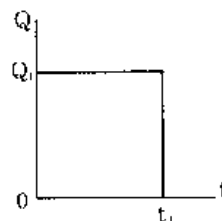


图 2.4 外力函数为矩形冲量时的情况

对方程(2.16)进行分步积分, 可得到下述结果:

$$x = \frac{Q_1}{k} \left[ 1 - e^{-nt} \left( \cos p_d t + \frac{n}{p_d} \sin p_d t \right) \right] \quad (0 \leq t \leq t_1) \quad (2.17)$$

如果略去阻尼, 即取  $n=0$  和  $p_d=p$ , 则方程(2.17)简化成:

$$x = \frac{Q_1}{k} (1 - \cos pt) \quad (0 \leq t \leq t_1) \quad (2.18)$$

在这种情况下, 如果考虑到  $t=0$  处的初始位移  $x_0$  和初始速度  $\dot{x}_0$ , 那么, 矩形脉冲  $Q_1$  作用于单自由度无阻尼系统的响应  $x$  为:

$$x = x_0 \cos pt + \frac{\dot{x}_0}{p} \sin pt + \frac{Q_1}{k} (1 - \cos pt) \quad (0 \leq t \leq t_1) \quad (2.19)$$

将此式对  $t$  求导, 相应的速度  $\dot{x}$  可由下式给出:

$$\frac{\dot{x}}{p} = -x_0 \sin pt + \frac{\dot{x}_0}{p} \cos pt + \frac{Q_1}{k} \sin pt \quad (0 \leq t \leq t_1) \quad (2.20)$$

式(2.19)和(2.20)就是求解这个问题要用到的基本公式。

(2) 计算方法和步骤

当外力  $Q$  为解析表达式时,可以直接利用杜哈米积分公式(2.12)进行积分求出系统的响应  $x$ 。当外力  $Q$  不能用解析式表达时,虽然也可以利用曲线拟合方法用一些近似表达式来代替,然后对这些近似表达式实施杜哈米积分,最后求出响应  $x$ 。然而,工程中求解响应的较为常用的方法是用一些简单的内插函数,采取一系列的重复计算来求出响应  $x$ 。下面介绍这种内插方法。

先假设图 2.1 的系统是一个有阻尼的单自由度系统。承受的外力  $Q$  随时间的变化如图 2.5 中的曲线所示。对于这样的外力函数,我们可以用一系列不同大小、具有不同的持续时间而又逐个相接的矩形脉冲来模拟,如图 2.5 中所示。第一个矩形脉冲的大小为  $Q_1$ ,持续时间为  $\Delta t_1 = t_1 - t_0$  (这里  $t_0 = 0$ );第二个矩形脉冲的大小为  $Q_2$ ,持续时间为  $\Delta t_2 = t_2 - t_1$  等等。在任一时间间隔  $t_{i-1} \leq t \leq t_i$  中,单自由度有阻尼系统的响应,可按时间  $t_{i-1}$  处初始条件的作用与间隔  $\Delta t_i$  里面这个冲量的作用之和来计算。由式(2.12)和式(2.7)可得:

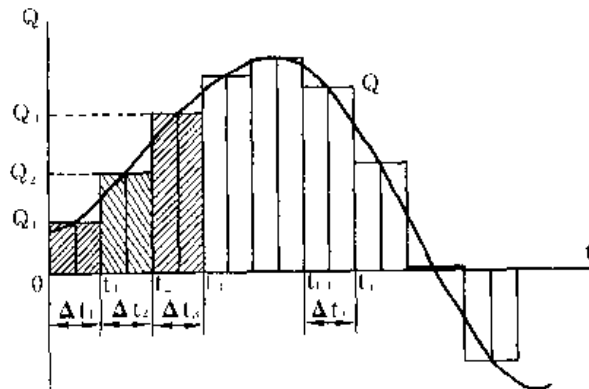


图 2.5 任意函数的分段常数内插方法示意图

$$x = e^{-n(t-t_{i-1})} \left[ x_{i-1} \cosh p_d(t-t_{i-1}) + \frac{\dot{x}_{i-1} + nx_{i-1}}{p_d} \sinh p_d(t-t_{i-1}) \right] + \frac{Q_i}{k} \left\{ 1 - e^{-n(t-t_{i-1})} \left[ \cosh p_d(t-t_{i-1}) + \frac{n}{p_d} \sinh p_d(t-t_{i-1}) \right] \right\} \quad (2.21)$$

在这个时间间隔的终了处,此式变成:

$$x_i = e^{-n\Delta t_i} \left( x_{i-1} \cosh p_d \Delta t_i + \frac{\dot{x}_{i-1} + nx_{i-1}}{p_d} \sinh p_d \Delta t_i \right) + \frac{Q_i}{k} \left[ 1 - e^{-n\Delta t_i} \left( \cosh p_d \Delta t_i + \frac{n}{p_d} \sinh p_d \Delta t_i \right) \right] \quad (2.22)$$

对于零阻尼的情况,方程(2.22)简化成:

$$x_i = x_{i-1} \cosh \Delta t_i + \frac{\dot{x}_{i-1}}{p} \sinh \Delta t_i + \frac{Q_i}{k} (1 - \cosh \Delta t_i) \quad (2.23)$$

这个时间间隔终了处的速度可由下式得出:

$$\frac{\dot{x}_i}{p} = -x_{i-1} \sinh \Delta t_i + \frac{\dot{x}_{i-1}}{p} \cosh \Delta t_i + \frac{Q_i}{k} \sinh \Delta t_i \quad (2.24)$$

方程(2.23)和(2.24)是计算第  $i$  步终了处的无阻尼响应,也就是第  $i+1$  步起始处的初始条件的两个递推公式。当第一步的起始条件给定之后,重复地应用这两个公式,就可

得到无阻尼单自由度系统在任意瞬间的位移和速度响应。为了得到良好的精确度,各个时间间隔  $\Delta t_i$  应当尽量地小,外力函数曲线下的面积误差应能近似地自行补偿,亦即图 2.5 中位于曲线上方的诸阴影线面积,应近似地等于曲线下方无阴影线的面积。为此,矩形脉冲的典型值  $Q_i$  应取为曲线上对应时间间隔  $\Delta t_i$  的中央处的纵坐标。在工程应用中,通常把各时间间隔  $\Delta t_i$  取成相等,而在每个小的时间间隔里,相应的内插函数值看成是一常数。这种内插方法称为分段常数内插方法。如果在每个小的时间间隔里,相应的内插函数值是线性增长的,那么这种内插方法称为分段线性内插方法。在这里,对后一种方法就不另作讨论了。

下面用式(2.23)和式(2.24)两个递推公式来求解线性系统对任意外力的响应。其计算步骤如下:

① 选择  $\Delta t_i = 0.1$  秒的均等时间步长,将图 2.2 上的曲线进行分段,如图 2.6 所示。每段上矩形冲量的值取成曲线上对应步长中间处的纵坐标,并将外力  $Q$  在各时间点的数值列成表 2.1。

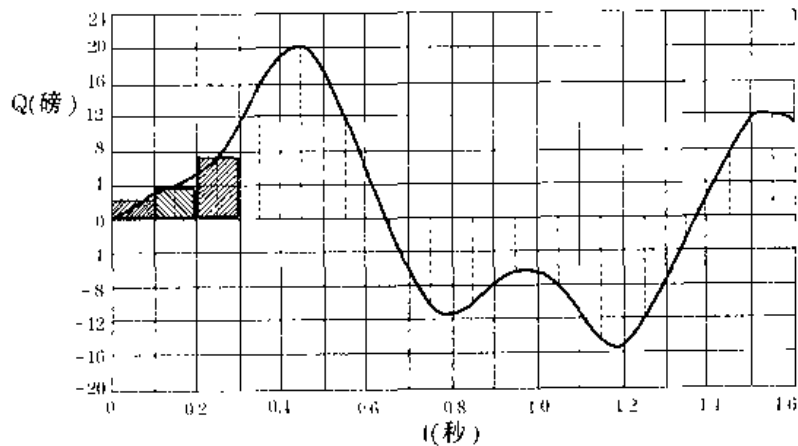


图 2.6 任意外力函数的分段常数内插方法

表 2.1 外力  $Q$  在各时刻的值

$\tau$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
$Q$	2.50	4.60	9.20	19.00	19.00	6.60	-9.20	-12.00
$\tau$	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6
$Q$	-8.40	-7.00	-12.70	-16.00	-9.30	1.50	11.20	11.20

② 无阻尼时,  $\tau = 2\pi/p$ , 已知  $\Delta t_i/\tau = \frac{1}{12}$ , 所以  $p\Delta t_i = \frac{\pi}{6}$ ,  $\cos p\Delta t_i = 0.866$ ,  $\sin p\Delta t_i = 0.500$ ,  $1 - \cos p\Delta t_i = 0.134$ 。

③ 开始时,  $x_0 = 0$ ,  $\dot{x}_0 = 0$ 。由  $i=0$  开始, 然后逐步计算出式(2.23)、(2.24)等号右端各项的数值, 最后求出每一步终止处的  $x_i$  和  $\dot{x}_i$ , 直至最后一步为止。

根据上面的计算过程, 可画出程序流程图如图 2.7

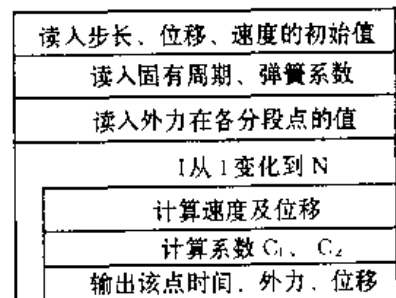


图 2.7 求解线性系统对任意外力响应的流程图

所示。还可绘制出外力的位移响应曲线。

## 2. 土方平衡调配计算

土方调配工作是土方施工设计的一个重要内容。土方调配的目的是在使土方总运输量最小或土方运输成本最小的条件下,确定填挖方区土方的调配方向和数量,从而达到缩短工期和降低成本的目的。例如图 2.8 为矩形广场,现已知各挖方和填方区的土方量和相互之间的平均运距,试求最优土方调配方案。

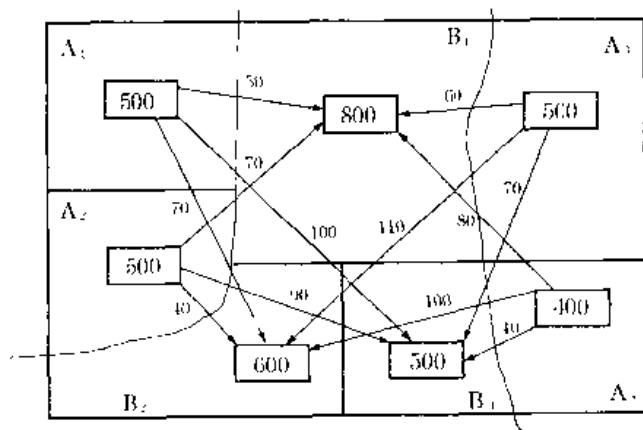


图 2.8 各调配区的土方量和平均运距

土方调配的方法一般采用“表上作业法”,现将方法简述如下:

### (1) 初始调配方案编制

先将图 2.8 中的数值标注在填、挖方平衡及运距表中,如表 2.2 所示。

表 2.2 填、挖方平衡及运距表

挖方区	填方区					
	B <sub>1</sub>		B <sub>2</sub>		B <sub>3</sub>	
A <sub>1</sub>		50		70		100
A <sub>2</sub>		70		40		90
A <sub>3</sub>		60		110		70
A <sub>4</sub>	×	80	×	100	(400)	40
填方量(m <sup>3</sup> )	800		600		500	
					1900	

而初始方案的编制采用“最小元素法”,即根据对应于最小的  $C_{ij}$  (平均运距) 取尽可能大的  $X_{ij}$  (土方调配数) 的原则进行调配。首先在运距表(小方格)中找一个最小数值,表 2.2 中  $C_{22}=C_{43}=40$  (任取其中一个,如  $C_{13}$ )。再确定  $X_{13}$  的值,使它尽可能大,即  $X_{13} = \min(400, 500) = 400$ 。由于 A<sub>4</sub> 挖方区的土方全部调到 B<sub>3</sub> 填方区,所以  $X_{11}$  和  $X_{12}$  都等于零。将 400 填入表 2.2 中的  $X_{43}$  格内,同时在  $X_{41}$ 、 $X_{42}$  格内画上一个“×”号。然后在没有填上数字和“×”号的方格内,再选一个运距最小的方格重复上述步骤,依次确定其余的  $X_{ij}$  数值,最后得出初始调配方案,如表 2.3 所示。

表 2.3 初始调配方案

挖方区	填方区						挖方量(m³)
	B <sub>1</sub>		B <sub>2</sub>		B <sub>3</sub>		
A <sub>1</sub>	(500)	50	×	70	×	100	500
A <sub>2</sub>	×	70	(500)	40	×	90	500
A <sub>3</sub>	(300)	60	(100)	110	(100)	70	500
A <sub>4</sub>	×	80	×	100	(400)	40	400
填方量(m³)	800		600		500		1900 1900

## (2) 最优方案的判别法

采用“位势法”判断初始方案是否为最优方案,该方法主要是通过求检验数  $\lambda_{ij}$  来判断,只要所有的检验数  $\lambda_{ij}$  大于 0,则该方案为最优方案;否则该方案不是最优方案,需要进行调整。下面介绍用“位势法”求检验数。

首先将初始方案中有调配数方格的平均运距  $C_{ij}$  列出来,然后按下式求出两组位势数  $u_i (i=1,2,\dots,m)$  和  $v_j (j=1,2,\dots,n)$ 。

$$C_{ij} = u_i + v_j \quad (2.25)$$

位势数求出后,便可根据下式计算各空格的检验数:

$$\lambda_{ij} = C_{ij} - u_i - v_j \quad (2.26)$$

现用“位势法”来判断表 2.3 中求得的初始方案是否是最优方案。首先把表 2.3 中有调配数方格的平均运距列于表 2.4 中。

表 2.4 有土方调配数的平均运距

挖方区 \ 填方区	位势	B <sub>1</sub>		B <sub>2</sub>		B <sub>3</sub>	
		$v_1 = 50$		$v_2 = 100$		$v_3 = 60$	
A <sub>1</sub>	$u_1 = 0$	0	50				
A <sub>2</sub>	$u_2 = -60$			0	40		
A <sub>3</sub>	$u_3 = 10$	0	60	0	110	0	70
A <sub>4</sub>	$u_4 = -20$					0	40

位势计算如下:

先令  $u_1 = 0$  则:

$$v_1 = C_{11} - u_1 = 50 - 0 = 50 \quad u_3 = C_{31} - v_1 = 60 - 50 = 10 \quad v_2 = C_{32} - u_3 = 110 - 10 = 100$$

$$v_3 = C_{33} - u_3 = 70 - 10 = 60 \quad u_2 = C_{22} - v_2 = 40 - 100 = -60 \quad u_4 = C_{43} - v_3 = 40 - 60 = -20$$

位势数求出后,在根据公式(2.26),依次求出各空格的检验数。如  $\lambda_{21} = C_{21} - u_2 - v_1 = 70 - (-60) - 50 = +80$ 。将计算结果填入表 2.5(在表 2.5 中只写“+”或“-”,可不必填入数字)。



表 2.5 检验数

挖方区 \ 填方区	位势	$B_1$		$B_2$		$B_3$	
		$v_1 = 50$		$v_2 = 100$		$v_3 = 60$	
$A_1$	$u_1 = 0$	0	50	-	70	+	100
$A_2$	$u_2 = -60$	+	70	0	40	+	90
$A_3$	$u_3 = 10$	0	60	0	110	0	70
$A_4$	$u_4 = -20$	+	80	+	100	0	40

表 2.5 出现了负的检验数,这说明初始方案不是最优方案,需要进一步进行调整。

### (3) 方案的调整

第一步:在所有负检验数中选一个(一般可选最小的一个),本例中便是  $C_{12}$ ,把它所对应的变量  $x_{12}$  作为调整对象。

第二步:找出  $x_{12}$  的闭回路。其作法是:从  $x_{12}$  格出发,沿水平与竖直方向前进,遇到适当的有数字(调配数)的方格作  $90^\circ$  转弯(也不一定转弯),然后继续前进,如果路线恰当,有限步后便能回到出发点,形成一条以有数字的方格为转弯点、用水平和竖直线联起来的闭回路,见表 2.6。

表 2.6 调配方案的调整

挖方区	填方区		
	$B_1$	$B_2$	$B_3$
$A_1$	500	$x_{12}$	
$A_2$		500	
$A_3$	300	100	100
$A_4$			400

第三步:从空格  $x_{12}$  出发,沿着闭回路(方向任意)一直前进,在各奇数次转角点的数字中挑出一个最小的(本例中便是在 500, 100 中选出 100);将 100 填入  $x_{12}$  方格中,被挑出的  $x_{12}$  为 0(该格变为空格);同时将闭回路上其他的奇数次转角上的数字都减去 100,偶数次转角上数字都增加 100,使得填挖方区的土方量仍然保持平衡。这样调整后,便得到表 2.7 的新调配方案。

对新调配方案,仍用“位势法”进行检验,看其是否已是最优方案。如果检验数中仍有负数出现,那就仍按上述步骤继续调整,直到找出最优方案为止。

表 2.7 新调配方案

挖方区 \ 填方区	位势	$B_1$		$B_2$		$B_3$		挖方量 ( $m^3$ )
		$v_1$	50	$v_2$	70	$v_3$	60	
$A_1$	$u_1 = 0$	400	50	100	70	+	100	500
$A_2$	$u_2 = -30$	+	70	500	40	-	90	500
$A_3$	$u_3 = 10$	400	60	+	110	100	70	500
$A_4$	$u_4 = -20$	+	80	+	100	400	40	400
土方量( $m^3$ )		300		600		500		1900

表 2.7 中所有检验数均为正号,故该方案即为最优方案。

该最优土方调配方案的土方总运输量为:

$$z = 400 \times 50 + 100 \times 70 + 500 \times 40 + 400 \times 60 + 100 \times 70 + 100 \times 40 = 94000 \text{ (} m^3 \cdot m \text{)}$$

最后将表 2.7 中的土方调配数值绘成土方调配图(图 2.9)。

本题的流程图见图 2.10。

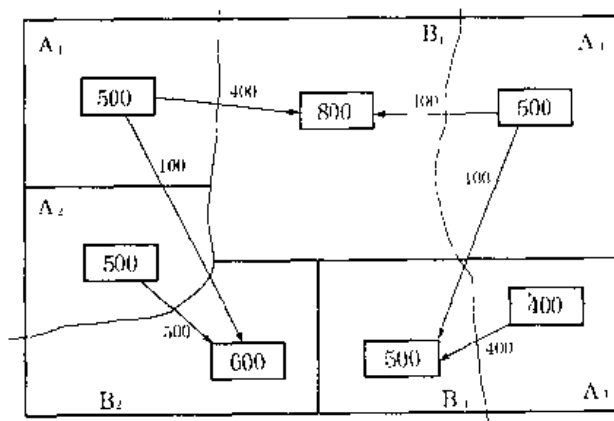


图 2.9 土方调配图

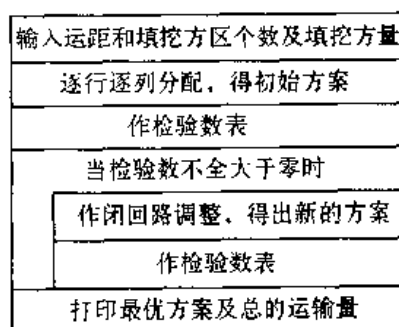


图 2.10 土方平衡调配计算流程图

## 第三章 常用算法设计

“如何编写程序”是学习高级语言程序设计的难点,也是学习的重点。初学者普遍感到头疼的是,碰到一个问题后不知从何下手,难以建立起明确的编程思路。针对这一普遍现象,本章根据教学基本要求,将常见的程序设计问题分为累加与累乘问题、数字问题、数值计算问题、数组的应用、子程序的应用和解不定方程等六类,分别总结每一类程序设计问题的思路,以引导读者掌握基本的程序设计方法和技巧。

### 3.1 累加与累乘问题

累加与累乘问题是很典型、最基本的一类算法,实际应用中很多问题都可以归结为累加与累乘问题。先看累加问题。

累加的数学递推式为:

$$S_0 = 0$$

$$S_i = S_{i-1} + X_i \quad (i=1,2,3,\dots)$$

其含义是第  $i$  次的累加和  $S$  等于第  $i-1$  次时的累加和  $S$  加上第  $i$  次时的累加项  $X$ 。从循环的角度讲,即是本次循环的  $S$  值等于上一次循环时的  $S$  值加上本次循环的  $X$  值,这可用下列赋值语句来实现:

$$S = S + X$$

显然,控制上述赋值语句重复执行若干次后, $S$  的值即若干个数之和。

特例 1 当  $X_i$  恒为 1 时,即  $S_i = S_{i-1} + 1$ ,  $S$  用于计数。

特例 2 当  $X_0 = 0$ , 且  $X_i = X_{i-1} + 1$  ( $i=1,2,3,\dots,N$ ) 时,  $S$  为  $1+2+3+\dots+N$  之值。

再看累乘问题。其数学递推式为:

$$P_0 = 1$$

$$P_i = P_{i-1} * X_i \quad (i=1,2,3,\dots)$$

其含义是第  $i$  次的累乘积  $P$  等于第  $i-1$  次时的累乘积  $P$  乘以第  $i$  次时的累乘项  $X$ 。从循环的角度讲,即是本次循环的  $P$  值等于上一次循环时的  $P$  值乘以本次循环的  $X$  值,这可用下列赋值语句来实现:

$$P = P * X$$

显然,控制上述赋值语句重复执行若干次后, $P$  的值即若干个数之积。

特例 1 当  $X_1 = X_2 = \dots = X_{N-1} = X_N = X$  时,  $P$  的值为  $X^N$ 。

特例2 当  $X_0 = 0$ , 且  $X_i = X_{i-1} + 1$  ( $i = 1, 2, 3, \dots, N$ ) 时,  $P$  的值为  $N!$ 。

递推问题常用迭代方法来处理, 即赋值语句  $S = S + X$  或  $P = P * X$  循环若干次。算法设计思路是:

(1) 写出循环体中需要重复执行的部分。这一部分要确定两个内容: 一是求每次要累加或累乘的数; 二是迭代关系  $S = S + X$  或  $P = P * X$ 。

(2) 确定终止循环的方式。一般有事先知道循环次数的计数循环和事先不知道循环次数的条件循环两种方式, 依具体情况而定。计数循环可用一个变量来计数, 当达到一定循环次数后即退出循环; 条件循环可根据具体情况确定一个循环的条件, 当循环条件不满足时即退出循环。

(3) 确定循环初始值, 即第一次循环时迭代变量的值。

(4) 重新检查, 以保证算法正确无误。

一般而言, 这一类问题的算法流程图基本框架如图 3.1 所示。

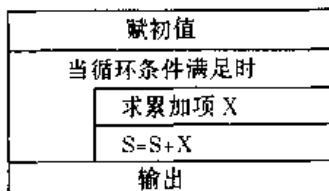


图 3.1 累加问题算法流程图基本框架

【例 3.1】已知  $S = \sum_{i=1}^N \frac{2}{(4i-3)(4i-1)}$ , 分别求:

- (1) 当  $N$  取 1000 时,  $S$  的值;
- (2)  $S < 0.78$  时的最大  $N$  值和与此时  $N$  值对应的  $S$  值;
- (3) 求  $S$  的值, 直到累加项小于  $10^{-4}$  为止。

第一种情况下, 属于循环次数已知的循环结构, 不难画出流程图如图 3.2(a) 所示。第二、三两种情况下, 属于循环次数未知的循环结构, 根据图 3.1 的流程图框架得到流程图分别如图 3.2(b) 和 (c) 所示。

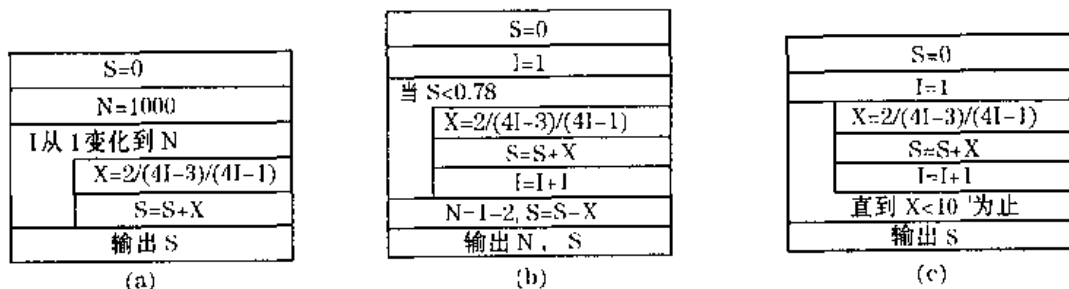


图 3.2 求  $S$  的流程图

根据流程图不难分别写出程序如下:

程序 1:

$S=0$

$N=1000$

```

DO I=1,N
  X=2.0/(4*I-3)/(4*I-1)
  S=S+X
END DO
PRINT *,"S=",S
END

```

运行结果如下:

S=7.852727E-01

程序 2:

```

S=0
I=1
DO WHILE(S<0.78)
  X=2.0/(4*I-3)/(4*I-1)
  S=S+X
  I=I+1
END DO
N=I-2
S=S-X
PRINT *,"N=",N,"S=",S
END

```

运行结果如下:

N= 23 S=7.799640E-01

程序 3:

```

S=0
I=1
X=2.0/(4*I-3)/(4*I-1)
DO WHILE(X>=1E-4)
  S=S+X
  I=I+1
  X=2.0/(4*I-3)/(4*I-1)
END DO
PRINT *,"S=",S
END

```

运行结果如下:

S=7.818270E-01

【例 3.2】 已知  $S = \sum_{i=1}^N \frac{\sin(x_i + y_i)}{1 + \sqrt{x_i y_i}}$

$$\text{其中 } x_i = \begin{cases} i & (i \text{ 为奇数}) \\ \frac{i}{2} & (i \text{ 为偶数}) \end{cases}, \quad y_i = \begin{cases} i^2 & (i \text{ 为奇数}) \\ i^3 & (i \text{ 为偶数}) \end{cases}.$$

从键盘输入 N 的值,求 S 的值。

程序如下:

S=0

READ \*,N

DO I=1,N

IF(MOD(I,2).EQ.1) THEN

X=I

Y=I \* \* 2

ELSE

X=I/2.0

Y=I \* \* 3

END IF

S=S+SIN(X+Y)/(1+SQRT(X\*Y))

END DO

PRINT \*, "S=", S

END

执行程序,从键盘输入:

30 ↵

输出结果为:

S=2.3750008

**【例 3.3】** 求  $y=f(1)+f(2)+\cdots+f(n)$

其中  $f(n)=(-1)^n \sqrt{2n^2+1}$ 。

(1)当  $n=50$  时,  $y$  的值是多少?

(2)当  $n=100$  时,  $y$  的值是多少?

程序如下:

Y=0

READ \*,N

DO I=1,N

F=(-1) \* \* I \* SQRT(2 \* I \* \* 2+1.0)

Y=Y+F

END DO

PRINT \*, "Y=", Y

END

当  $n=50$  时,  $y$  的值是 35.145430。

当  $n=100$  时,  $y$  的值是 70.499030。

【例 3.4】 输入  $x$  值,按下列算式计算  $\cos x$ :

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$$

直到最后一项的绝对值小于  $10^{-5}$  为止。

程序如下:

```
Y=1.0
READ *,X
I=1
F=1.0
DO WHILE(ABS(F)>=1E-5)
    F=F*(-1)*X**2/(2*I*(2*I-1))
    Y=Y+F
    I=I+1
END DO
PRINT *, "Y=",Y
END
```

【例 3.5】 计算两个 4 位整数  $M$  和  $N$  的乘积,该乘积分别存放于两个整型变量  $H$  和  $L$  中,其中  $H$  存放乘积的高 4 位, $L$  存放乘积的低 4 位。例如,若  $M=N=9999$ ,则  $M*N=9999*9999=99980001$ ,乘积前 4 位数 9998 存入  $H$ ,后 4 位 0001 存放  $L$ 。

显然这属于累加问题, $M*N$  等于  $N$  个  $M$  相加。程序如下:

```
INTEGER M,N,H,L,I
READ(*,*) M,N
H=0
L=0
DO I=1,N
    L=L+M
    IF (L.GE.10000) THEN
        L=L-10000
        H=H+1
    END IF
END DO
WRITE(*,*) 'H=',H,'L=',L
END
```

## 3.2 数字问题

数字问题主要研究整数的一些自身性质与相互关系。处理过程中常常要用到求余

数、分离数字及判断整除等技巧,务必熟练掌握。例如:

(1) 判断一个整数  $M$  能否被另一个整数  $N$  整除

方法 1: 若  $\text{MOD}(M, N)$  的值为 0, 则  $M$  能被  $N$  整除, 否则不能。

方法 2: 若  $M - \text{INT}(M/N) * N$  的值为 0, 则  $M$  能被  $N$  整除, 否则不能。

事实上,  $\text{MOD}(M, N)$  的函数值是  $M$  除以  $N$  的余数, 正是按照表达式  $M - \text{INT}(M/N) * N$  得到的计算结果。  $M - \text{INT}(M/N) * N$  也可以写成  $M - M/N * N$ 。

(2) 分离自然数  $M$  各位的数字

根据  $\text{MOD}$  函数的定义,  $\text{MOD}(M, 10)$  的值是  $M$  的个位数字,  $\text{MOD}(M/10, 10)$  的值是  $M$  的十位数字, 依此类推, 可以得到  $M$  的更高位数字。

数字问题的提法往往是求某一范围内符合某种条件的数。这一类问题的算法设计思路是:

(1) 考虑判断一个数是否满足条件的算法。有时侯可以直接用一个关系表达式或逻辑表达式来判断, 如判断奇数、偶数。但更多的情况是无法直接用一个条件表达式来判断, 这时可根据定义利用一个循环结构来进行判断。

(2) 在指定范围内重复执行“判断一个数是否满足条件”的程序段, 从而求得指定范围内全部符合条件的数。这里用的方法是穷举。

一般而言, 这一类问题的算法流程图基本框架如图 3.3 所示。

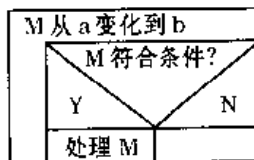


图 3.3 数字问题算法流程图基本框架

**【例 3.6】** 求所有各位数字的立方和等于 1099 的三位整数。

按照图 3.3, 不难写出程序如下:

```
INTEGER M, M1, M2, M3
```

```
DO M=100,999
```

```
    M1=MOD(M,10)**3
```

```
    M2=MOD(M/10,10)**3
```

```
    M3=(M/100)**3
```

```
    IF (M1+M2+M3.EQ.1099) PRINT *,M
```

```
END DO
```

```
END
```

运行结果是:

```
379
```

```
397
```

```
739
```

```
793
```

```
937
```

```
973
```

**【例 3.7】** 考察[1000,2000]范围内的全部素数。



- (1) 最小的素数;
- (2) 由小到大第 100 个素数;
- (3) 全部素数之和。

程序如下:

```
INTEGER(4) S
K=0
S=0
DO N=1001,1999,2
    M=SQRT(N*1.0)
    I=2
    DO WHILE(MOD(N,I) /= 0. AND. I <= M)
        I=I+1
    END DO
    IF (I > M) THEN
        K=K+1
        IF(K==1) PRINT *, "最小的素数为:", N
        IF(K==100) PRINT *, "由小到大第 100 个素数为:", N
        S=S+N
    END IF
END DO
PRINT *, "全部素数之和为:", S
END
```

运行结果是:

最小的素数为: 1009

由小到大第 100 个素数为: 1721

全部素数之和为: 200923

**【例 3.8】** 若两个素数之差是 2, 则称这两个素数是一对孪生数。例如, 3 和 5 是一对孪生数。求 [2, 500] 区间内:

- (1) 孪生数的对数;
- (2) 最大的一对孪生数。

程序如下:

```
INTEGER TWIN
LOGICAL SU
DO N=3,497,2
    IF(SU(N). AND. SU(N+2)) THEN
        K=K+1
        TWIN=N
    END IF
```

```

END DO
PRINT *, "孪生数的对数:", K
PRINT *, "最大的一对孪生数:", TWIN, TWIN + 2
END
LOGICAL FUNCTION SU(N)
    SU = .FALSE.
    M = SQRT(N * 1.0)
    I = 2
    DO WHILE (MOD(N, I) /= 0. AND. I <= M)
        I = I + 1
    END DO
    IF (I > M) SU = .TRUE.
END

```

输出结果是:

孪生数的对数:        24

最大的一对孪生数:        461        463

**【例 3.9】** 若正整数  $N$  的所有因子之和等于  $N$  的倍数, 则称  $N$  为红玫瑰数。如 28 的因子之和为  $1+2+4+7+14+28=56=28*2$ , 故 28 是红玫瑰数。求:

- (1)  $[1, 700]$  之间最大的红玫瑰数;
- (2)  $[1, 700]$  之间有多少个红玫瑰数。

程序如下:

```

INTEGER ROSE, SUM
DO N=1, 700
    SUM=0
    DO J=1, N
        IF (MOD(N, J) == 0) SUM=SUM+J
    END DO
    IF (MOD(SUM, N) == 0) THEN
        K=K+1
        ROSE=N
    END IF
END DO
PRINT *, "最大的红玫瑰数:", ROSE
PRINT *, "红玫瑰数的个数:", K
END

```

输出结果是:

最大的红玫瑰数:        672

红玫瑰数的个数:        6

【例 3.10】 求[2,1000]范围内因子(包括 1 和该数本身)个数最多的数及其因子个数。

程序如下:

```
INTEGER::G_MAX=0,MAX=0
DO N=2,1000
    K=0
    DO J=1,N
        IF(MOD(N,J)==0) K=K+1
    END DO
    IF(MAX<K) THEN
        MAX=K
        G_MAX=N
    END IF
END DO
PRINT *, "因子个数最多的数:", G_MAX
PRINT *, "因子个数:", MAX
END
```

运行程序,结果如下:

因子个数最多的数: 840

因子个数: 32

【例 3.11】 若某不含数字 0 的三位正整数,其平方数至少有一位同样的数字,则称该三位数为三重数。例如,由于  $511^2=261121$ (有三位 1),所以 511 为三重数。求出:

- (1) 按升序排列第 10 个三重数;
- (2) 按升序排列前 10 个三重数之和;
- (3) 所有三重数之和。

程序如下:

```
INTEGER(4) M,SUM,TOTAL,NUM,I,J,K
INTEGER A(6)
SUM=0
TOTAL=0
NUM=0
DO N=100,999
    IF(MOD(N/10,10)/=0.AND.MOD(N,10)/=0) THEN
        M=N*N
        J=0
        DO WHILE (M/=0)
            J=J+1
            A(J)=MOD(M,10)
            M=M/10
        END DO
        SUM=SUM+A(J)
        TOTAL=TOTAL+N
        NUM=NUM+1
    END IF
END DO
```

```

END DO
K=J
DO J=1,K-2
    L=1
    DO I=J+1,K
        IF(A(J)==A(I)) L=L+1
    END DO
    IF(L>=3) THEN
        NUM=NUM+1
        IF(NUM==10) PRINT *,"第 10 个三重数:",N
        IF(NUM<=10) SUM=SUM+N
        TOTAL=TOTAL+N
        EXIT
    END IF
END DO
END IF
END DO
PRINT *,"前 10 个三重数之和:",SUM
PRINT *,"所有三重数之和:",TOTAL
END

```

输出结果为:

```

第 10 个三重数:      258
前 10 个三重数之和:    1826
所有三重数之和:      60473

```

### 3.3 数值计算问题

数值计算是“计算方法”课程研究的对象,主要研究如何用计算机来求一些数学问题的数值解。目前数值计算方法已日臻成熟,许多问题都有了现成的算法或软件包。详细内容可参阅数值分析或计算方法方面的专著,或直接使用有关科学计算软件,如 MATLAB。

**【例 3.12】** 用牛顿迭代法求方程  $f(x)=0$  在  $x=x_0$  附近的实根。直到  $|x_n - x_{n-1}| \leq \epsilon$  为止。

牛顿迭代公式为:  $x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}$

本质上讲,这属于递推问题,采用迭代方法不难得到图 3.4 所示的算法。

N=0	
输入 X	
	F=F(X)
	F1=F'(X)
	X=X-F/F1
	N=N+1
直到 F/F1 ≤ε 为止	
输出 N 和 X	

图 3.4 牛顿迭代法求方程的根

设  $f(x) = x^2 - a$ , 则迭代公式为:

$$x_n = \frac{1}{2} \left( x_{n-1} + \frac{a}{x_{n-1}} \right)$$

显然此时方程的根即  $\sqrt{a}$ , 利用此迭代公式可以求  $\sqrt{a}$  的近似值。假定取  $x_0 = a/2$ ,  $\epsilon = 10^{-4}$ , 程序如下:

```

REAL A,X
N=0
PRINT *, "INPUT A: "
READ *, A
X=A/2
X1=(X+A/X)/2
DO WHILE(ABS(X1-X) >= 1E-4)
    N=N+1
    X=X1
    X1=(X+A/X)/2
END DO
PRINT *, X
END

```

【例 3.13】 求  $S = \int_a^b f(x) dx$  之值。

算法一: 矩形法。根据定积分的几何意义, 将积分区间  $[a, b]$   $n$  等分,  $n$  个小的曲边梯形面积之和即定积分的近似值。矩形法用小矩形代替小曲边梯形, 求出各小矩形面积, 然后累加之, 所以本质上讲这是一个累加问题, 算法如图 3.5 所示。

也可以先找出求几个小矩形面积之和的公式, 然后根据公式编写程序。

$$S = S_1 + S_2 + \cdots + S_n$$

$$= h \cdot f(a) + h \cdot f(a+h) + \cdots + h \cdot f[a+(n-1)h]$$

$$= h \sum_{i=1}^n f[a+(i-1)h]$$

输入 A, B, N	
X=A	
H=(B-A)/N	
F=F(X)	
S=0	
I 从 1 变化到 N	
	SI=F*H
	S=S+SI
	X=X+H
	F=F(X)
输出 S	

图 3.5 矩形法求定积分

其中  $h = \frac{b-a}{n}$ 。

显然这是一个累加问题, 不难设计算法。

当  $n=100$  时, 求  $s = \int_{-1}^1 \sqrt{1-x^2} dx$  的程序如下:

```
F(X)=SQRT(1-X*X)
PRINT *, "INPUT A,B,N: "
READ *, A,B,N
H=(B-A)/N
S=0
DO I=1,N
    S=S+H * F(A+(I-1)*H)
END DO
PRINT *, S
END
```

运行程序并输入 -1,1,100, 结果如下:

1.569134

算法二: 梯形法。梯形法用小梯形代替小曲边梯形。

第一个小梯形的面积为:  $S_1 = \frac{f(a+h) + f(a)}{2} \cdot h$

第二个小梯形的面积为:  $S_2 = \frac{f(a+2h) + f(a+h)}{2} \cdot h$

.....

第  $i$  个小梯形的面积为:  $S_i = \frac{f(a+ih) + f[a+(i-1)h]}{2} \cdot h$

.....

第  $n$  个小梯形的面积为:  $S_n = \frac{f[a+(n-1)h] + f(b)}{2} \cdot h$

本质上讲这也是一个累加问题。

也可以先找出求几个小梯形面积代数公式, 然后根据此式设计算法。

$$S = S_1 + S_2 + \cdots + S_n = h \cdot \frac{f(a) + f(b)}{2} + h \cdot \sum_{i=1}^{n-1} f(a+ih)$$

根据上式, 当  $n=100$  时, 求  $s = \int_{-1}^1 \sqrt{1-x^2} dx$  的程序如下:

```
F(X)=SQRT(1-X*X)
PRINT *, "INPUT A,B,N: "
READ *, A,B,N
H=(B-A)/N
S=H * (F(A)+F(B))/2
DO I=1,N-1
```

```
S=S+H * F(A+I * H)
```

```
END DO
```

```
PRINT *,S
```

```
END
```

运行程序并输入-1,1,100,结果如下:

1.569134

算法三:定步长辛普生法。辛普生法在一小区间内用一抛物线代替原来的曲线  $f(x)$ ,其公式为:

$$\int_a^b f(x)dx \approx \frac{h}{3} \left\{ f(a) + f(b) + 2 \sum_{i=1}^{n-1} f(a + 2ih) + 4 \sum_{i=1}^n f[a + (2i-1)h] \right\}$$

其中: $n$  为小区间个数,  $h = \frac{b-a}{2n}$ 。

为了计算方便起见,把上式改写为:

$$\int_a^b f(x)dx \approx \frac{b-a}{3n} \left\{ 0.5[f(a) + f(b)] + \sum_{i=1}^n [2f(a + (2i-1)h) + f(a + 2ih)] \right\}$$

子程序 SIMP 是用定步长辛普生法求定积分,其中虚参 A,B 代表积分下限和上限,N 为由积分精度要求确定的小区间个数,S 为积分结果。

函数子程序为被积函数值。假定计算定积分:

$$s = \int_{-1}^1 \sqrt{1-x^2} dx$$

程序如下:

```
PROGRAM PIO
```

```
N=100
```

```
CALL SIMP(-1.0,1.0,N,S)
```

```
WRITE(*,10)S
```

```
10 FORMAT(1X,2HS=,F10.6)
```

```
END
```

```
FUNCTION F(X)
```

```
F=SQRT(1-X * X)
```

```
END
```

```
SUBROUTINE SIMP(A,B,N,S)
```

```
H=(B-A)/(2 * N)
```

```
S=0.5 * (F(A)-F(B))
```

```
DO I=1,N
```

```
    S1=F(A+(2 * I-1) * H)
```

```
    S2=F(A+2 * I * H)
```

```
    S=S+2.0 * S1+S2
```

```

END DO
S=(B-A)*S/(3*N)
END
运行程序,结果如下:
S=      1.570569

```

### 3.4 数组的应用

数组是最常用的结构数据类型之一,常用它来实现多个相同类型数据的存储和处理。绝大多数高级语言都提供了数组类型,在有些语言中这甚至是唯一的结构类型。

【例 3.14】 已知

$$\begin{cases} F_0 = F_1 = 0 \\ F_2 = 1 \\ F_n = F_{n-1} - 2F_{n-2} + F_{n-3} \quad (n > 2) \end{cases}$$

在  $F_0$  到  $F_{100}$  中:

- (1) 负数的个数;
- (2) 1888 是第几项(约定  $F_0$  是第 0 项,  $F_1$  是第 1 项)。

利用数组,编写程序如下:

```

INTEGER F(0:100)
F(0)=0
F(1)=0
F(2)=1
K=0
DO N=3,100
    F(N)=F(N-1)-2*F(N-2)+F(N-3)
    IF(F(N)<0) K=K+1
    IF(F(N)==1888) PRINT *, "1888 是第", N, "项"
END DO
PRINT *, "负数的个数:", K
END

```

运行程序,结果如下:

```

1888 是第      29 项
负数的个数:      47

```

【例 3.15】 因子和等于它本身的数称为完数。如 6 的因子是 1,2,3 即  $1+2+3=6$ , 所以 6 是完数。求 2~1000 中的完数并输出其全部因子。

输出格式为(以 6 为例): $6=1+2+3$ 。



程序如下:

```

INTEGER F(20)
INTEGER SUM
DO N=2,1000
    SUM=0
    K=0
    DO I=1,N/2
        IF(MOD(N,I)=-1) THEN
            K=K+1
            SUM=SUM+I
            F(K)=I
        END IF
    END DO
    IF(SUM==N) THEN
        WRITE(*,100) N,"=",F(1),(" + ",F(I),I=2,K)
    END IF
END DO
100 FORMAT(1X,I5,A1,I4,20(A1,I4))
END

```

运行程序,结果如下:

```

6= 1+ 2+ 3
28= 1+ 2+ 4+ 7+ 14
496= 1+ 2+ 4+ 8+ 16+ 31+ 62+ 124+ 248

```

【例 3.16】 Josephus 问题。有  $M$  个同学围成一个圆圈做游戏,从某人开始编号(编号为  $1 \sim M$ ),并从 1 号同学开始报数,数到  $N$  的同学被取消游戏资格,下一个同学(第  $N+1$  个)又从 1 开始报数,数到  $N$  的同学便第二个被取消游戏资格,如此重复,直到最后一个同学被取消游戏资格,求依次被取消游戏资格的同学编号。

分析:定义一个一维动态数组  $K$ ,各元素的下标代表  $M$  个同学的编号,各元素的值代表同学是否被取消游戏资格,以 1 表示未被取消,以 0 表示已被取消,这样做的好处是,在对同学报数作统计时,可以直接累加  $K$  数组元素的值。当  $K$  数组元素的值全为 0 时,游戏结束,算法如图 3.6 所示。

程序如下:

```

INTEGER, DIMENSION(:), ALLOCATABLE :: K

```

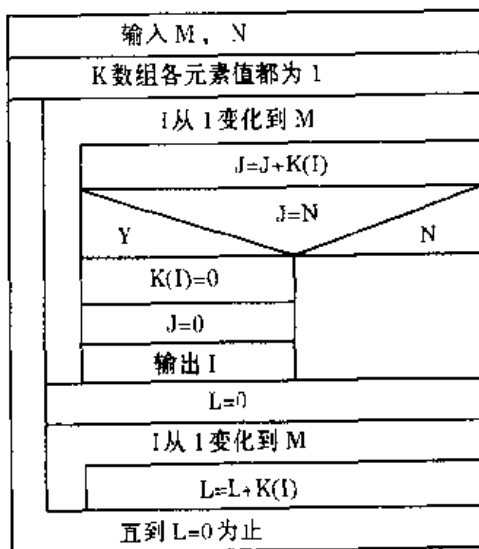


图 3.6 Josephus 问题的算法

```
READ *,M,N
ALLOCATE(K(M))
K=1
L=M
J=0
DO WHILE(L>0)
DO I=1,M
    J=J+K(I)
    IF(J==N) THEN
        K(I)=0
        J=0
        PRINT *,I
    END IF
END DO
L=0
DO I=1,M
    L=L+K(I)
END DO
END DO
DEALLOCATE(K)
END
```

也可以设一个变量来统计被取消游戏资格的同学的个数,当变量值等于 M 时,游戏结束,请按此思路改写程序。

**【例 3.17】** 有一篇文章,包括若干行,每行有 40 个字符,要求统计全文中大写字母 A~Z 出现的次数。

分析:这是一个分类统计问题,容易想到的算法是采用多分支选择结构来统计出不同类别数据的个数,但当类别很多时,这样做是十分烦琐的。较为简便的采用数组作分类统计,首先根据不同的类别来找到分类号,然后以分类号作为数组的下标,采取按分类号对号入座的方法,从而省去条件判断。

考虑到 26 个字母在 ASCII 码表中是连续排列的,求任一字母 CH 所对应的分类号 K 的表达式可以写成:

$K = \text{CH 的 ASCII 码} - \text{字母 A 的 ASCII 码} + 1$

显然 CH 等于 A 时, K 的值为 1, CH 等于 B 时, K 的值为 2, ..., CH 等于 Z 时, K 的值为 26。用 NUM 数组来作分类统计,下标变量 NUM(1), NUM(2), ..., NUM(26) 分别统计字母 A, B, ..., Z 的个数。

程序如下:

```
CHARACTER STR * 40, CH * 1
INTEGER, :: NUM(26) = 0
```

```

OPEN(8,FILE="EXAMPLE.TXT")
DO WHILE(.NOT.EOF(8))
    READ(8,*) STR
    DO I=1,40
        CH=STR(I:I)
        K=ICHAR(CH)-ICHAR("A")+1
        NUM(K)=NUM(K)+1
    END DO
END DO
DO I=1,26
    PRINT 100,"字符",CHAR(ICHAR("A")+I-1),"出现的次数:",NUM(I)
END DO
100 FORMAT(1X,A4,A1,A12,I3)
END

```

【例 3.18】 采用变化的冒泡排序法将  $n$  个数按从大到小的顺序排列：对  $n$  个数，从第一个直到第  $n$  个，逐次比较相邻的两个数，大者放前面，小者放后面，这样得到的第  $n$  个数是最小的，然后对前面  $n-1$  个数，从第  $n-1$  个到第 1 个，逐次比较相邻的两个数，大者放前面，小者放后面，这样得到的第 1 个数是最大的。对余下的  $n-2$  个数重复上述过程，直到按从大到小的顺序排列完毕。

程序如下：

```

PARAMETER (N=10)
INTEGER A(N),I,HIGH,TEMP
READ*,A
LOW=1
HIGH=N
DO WHILE(LOW<HIGH)
    DO I=LOW,HIGH-1
        IF(A(I)<A(I+1)) THEN
            TEMP=A(I)
            A(I)=A(I+1)
            A(I+1)=TEMP
        END IF
    END DO
    HIGH=HIGH-1
    IF(LOW<HIGH) THEN
        DO I=HIGH,LOW+1,-1
            IF(A(I)>A(I-1)) THEN
                TEMP=A(I)

```

```

        A(I)=A(I-1)
        A(I-1)=TEMP
    END IF
END DO
LOW=LOW+1
END IF
END DO
WRITE(*,'(1X,5I5)')A
END

```

### 3.5 子程序的应用

子程序是将反复要用到的程序段定义为一个独立的程序单元,当需要的时候,不需重复书写而直接调用即可。利用子程序可实现应用程序的模块化结构。

【例 3.19】 已知  $y = \frac{f(40)}{f(30)+f(20)}$

(1) 当  $f(n) = n + 10 \ln(n^2 + 5)$  时,  $y$  的值是多少;

(2) 当  $f(n) = 1 \times 2 + 2 \times 3 + 3 \times 4 + \cdots + n \times (n+1)$  时,  $y$  的值是多少。

第(1)题可以用语句函数完成。程序如下:

```

F(N)=N+10*LOG(N*N+5,0)
Y=F(40)/(F(30)+F(20))
PRINT*,Y
END

```

运行程序,结果如下:

6.389509E-1

题(1)也可以用函数子程序完成。程序如下:

```

Y=F(40)/(F(30)+F(20))
PRINT*,Y
END
FUNCTION F(N)
F=N+10*LOG(N*N+5,0)
END

```

第(2)题可以用函数子程序完成。程序如下:

```

Y=F(40)/(F(30)+F(20))
PRINT*,Y
END

```

```

FUNCTION F(N)
F=0
DO I=1,N
    F=F+I*(I+1)
END DO
END

```

运行程序,结果如下:

1.766154

第(2)题也可以用子例行程序完成。程序如下:

```

CALL SUB(40,F1)
CALL SUB(30,F2)
CALL SUB(20,F3)
Y=F1/(F2+F3)
PRINT *,Y
END
SUBROUTINE SUB(N,F)
F=0
DO I=1,N
    F=F+I*(I+1)
END DO
END

```

**【例 3.20】** 一个数为素数,且依次从低位去掉 1 位,2 位,……,所得的各数仍都是素数,则称该数为超级素数,例如 239。试求[100,9999]之内:

- (1) 超级素数的个数;
- (2) 所有超级素数之和;
- (3) 最大的超级素数。

程序如下:

```

LOGICAL SU
INTEGER,; NUM=0,SUM=0,MAX
DO N=100,9999
    IF (N>999) THEN
        IF (SU(N).AND.SU(N/10).AND.SU(N/100).AND.&
            SU(N/1000)) THEN
            NUM=NUM+1
            SUM=SUM+N
            MAX=N
        END IF
    ELSE

```

```

    IF (SU(N). AND. SU(N/10). AND. SU(N/100)) THEN
    NUM=NUM+1
    SUM=SUM+N
    MAX=N
    END IF
  END IF
END DO
PRINT *, "超级素数的个数:", NUM
PRINT *, "所有超级素数之和:", SUM
PRINT *, "最大的超级素数:", MAX
END
LOGICAL FUNCTION SU(N)
SU=.FALSE.
I=2
M=SQRT(N*1.0)
DO WHILE(MOD(N,I)/=0. AND. I<=M)
  I=I+1
END DO
IF(I>M. AND. N>1) SU=.TRUE.
END

```

运行程序,结果如下:

```

超级素数的个数:    30
所有超级素数之和:   75548
最大的超级素数:    7393

```

**【例 3.21】** 寻求并输出 3000 以内的亲密数对。亲密数对的定义为:若正整数 A 的所有因子(不包括 A)之和为 B, B 的所有因子(不包括 B)之和为 A, 且  $A \neq B$ , 则称 A 与 B 为亲密数对。

由于“求一正整数的所有因子之和”不只一次用到, 宜用过程或子程序表示。程序如下:

```

DO I=2,3000
  CALL FSUM(I,J)
  CALL FSUM(J,K)
  IF ((I.EQ.K). AND. (I.NE.J)) THEN
    WRITE(*,20)I,J
  END IF
END DO
20 FORMAT(5X,2I8)
END

```

```

SUBROUTINE FSUM(L,M)
M=1
DO N=2,L/2
  IF (MOD(L,N).EQ.0) THEN
    M=M+N
  END IF
END DO
END

```

【例 3.22】 设有一个游泳池如图 3.7 所示(单位:米)。已知步行速度为 2 米每秒,游泳速度为 1.2 米每秒。求 A 点到 B 点的最短时间和相应的路线(即 A 点步行到 X,再下水游泳到 B 点)。所谓的“路线”是指求出的 A 点到 X 点的距离,误差为 0.1 米。

分析:定义语句函数 F1,计算并返回 X 点到 B 点的距离:  $\sqrt{(40-x)^2+30^2}$  (米)。再定义函数子程序 F2,计算并返回 A 点步行经过 X 点游泳到 B 点的时间:  $\frac{x}{2} + \frac{F1 \text{ 函数返回值}}{1.2}$  (秒)。最后编写主程序,调用 F2 函数子程序,求 A 点到 B 点的最短时间和相应的路线。

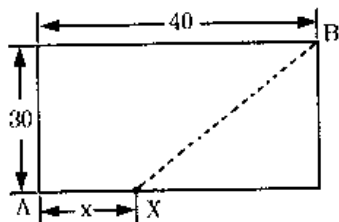


图 3.7 游泳池示意图

程序如下:

```

FUNCTION F2(X)
F1(X)=SQRT((40-X)**2+30**2)
F2=X/2+F1(X)/1.2
END
REAL MIN_X,MIN_T
MIN_X=0.0
MIN_T=40.0/2+30/1.2
DO X=0.1,40.0,0.1
  T=F2(X)
  IF(T<MIN_T) THEN
    MIN_T=T
    MIN_X=X
  END IF
END DO

```

```
PRINT *, "A 点到 B 点的最短时间为", MIN_T, "秒"
PRINT *, "A 点到 X 点的距离为", MIN_X, "米"
END
```

运行结果是:

A 点到 B 点的最短时间为 40.000000 秒

A 点到 X 点的距离为 17.500000 米

### 3.6 解不定方程

方程的个数少于未知数的个数称为不定方程,这类方程没有唯一解,而有多组解。对于这类问题无法用解析法解,只能将所有可能的解一个一个地去试,看是否满足方程,如满足就是方程的解。这里用的方法是穷举法。穷举对于计算机来说是很方便的。

**【例 3.23】** 求不定方程组

$$\begin{cases} x+y+z=20 \\ 25x+20y+13z=400 \end{cases}$$

的全部自然数解。

程序如下:

```
INTEGER X,Y,Z
DO X=1,18
  DO Y=1,18
    Z=20-X-Y
    IF(25 * X+20 * Y+13 * Z==400) THEN
      PRINT *,X,Y,Z
    END IF
  END DO
END DO
END
```

输出结果为:

7      8      5

**【例 3.24】** 求满足

$$\begin{cases} A \cdot B=716699 \\ A+B \text{ 最小} \end{cases}$$

的 A 和 B。

程序如下:

```
INTEGER A,B,S_MIN,A_MIN,B_MIN
S_MIN=716699
```



```

DO A=1,SQRT(716699.0),2
  TEMP=716699.0/A
  IF(TEMP==INT(TEMP)) THEN
    B=TEMP
    IF(S_MIN>A+B) THEN
      S_MIN=A+B
      A_MIN=A
      B_MIN=B
    END IF
  END IF
END DO
PRINT *,A_MIN,B_MIN
END

```

输出结果为:  
563 1273

### 思考题

1. 求  $Z = \sum_{i=1}^N (X_i - Y_i)^2$  的值(N 的值从键盘输入)。

$$\text{其中 } X = \begin{cases} i & (i \text{ 为奇数}) \\ \frac{i}{2} & (i \text{ 为偶数}) \end{cases}, \quad Y = \begin{cases} i^2 & (i \text{ 为奇数}) \\ i^3 & (i \text{ 为偶数}) \end{cases}.$$

(1) 当 N 取 10 时,求 Y 的值;

答案:1304735

(2) 当 N 取 15 时,求 Y 的值。

答案:11841724

2. 已知  $y = 1 - \frac{1}{2} + \frac{1}{4} - \cdots + \frac{1}{2n}$

(1) 求  $y > 4$  时的最小 n 值。

答案:227

(2) 与(1)的 n 值对应的 y 值。

答案:4.002183

3. 已知  $y = \frac{e^{0.3x} - e^{-0.3x}}{2} \cdot \sin(x + 0.3)$

(1) 当 x 取 -2.0 时,求 y 的值。

答案:0.6313470

(2) 当 x 取 -3.0, -2.9, -2.8, ..., 2.9, 3.0 时,求各点 y 值之和。

答案:19.000540

4. 已知  $e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{N!}$ , 若 x 取 0.5, N 取 100, 则  $e^x$  的值是多少?

答案:1.648721

5. 求  $S_n = a + aa + aaa + \cdots + \underbrace{aa \cdots a}_{n \uparrow a}$  的值。其中 a 为 1~9 之间的一个整数。

(提示:累加项的递推关系为  $x_n = x_{n-1} * 10 + a$ )

6. 根据下式求  $\pi$  的近似值。

$$\arctan x = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots \quad ①$$

$$\text{因为 } \tan \frac{\pi}{4} = 1, \text{ 所以 } \pi = 4 \cdot \arctan 1 \quad ②$$

$$\text{因为 } \tan \frac{\pi}{6} = \frac{1}{\sqrt{3}}, \text{ 所以 } \pi = 6 \cdot \arctan \frac{1}{\sqrt{3}} \quad ③$$

(1) 联合①②或①③, 求  $\pi$  的近似值(精度为  $10^{-7}$ )。

(2) ①②好, 还是①③好? 为什么?

7. 求 585 素数因子的个数及全部因子之和。 答案: 3, 506

8. 若一个正整数有偶数个不同的真因子, 则称该数为幸运数。如 4 有 2 个真因子 1 和 2, 故 4 是幸运数。求  $[2, 100]$  之间全部幸运数之和。 答案: 384

9. 若两个连续自然数的乘积减 1 是素数, 则称这两个连续自然数是亲密数对, 该素数是亲密素数。例如,  $2 \times 3 - 1 = 5$ , 由于 5 是素数, 所以 2 和 3 是亲密数对, 5 是亲密素数。求  $[2, 50]$  区间内:

(1) 亲密数对的对数。 答案: 28

(2) 与上述亲密数对对应的所有亲密素数之和。 答案: 21066

10. 已知 24 有 8 个因子: 1, 2, 3, 4, 6, 8, 12, 24, 而 24 正好能被 8 整除, 求  $[1, 100]$  之间:

(1) 有多少个整数能被其因子的个数整除? 答案: 16

(2) 符合(1)的最大整数。 答案: 96

(3) 符合(1)的所有整数之和。 答案: 686

11. 梅森尼数是指能使  $2^n - 1$  为素数的数  $n$ , 求  $[1, 21]$  内:

(1) 有多少个梅森尼数。 答案: 7

(2) 最大的梅森尼数。 答案: 19

(3) 第二大的梅森尼数。 答案: 17

12. 倒勾股数是满足公式:

$$\frac{1}{A^2} + \frac{1}{B^2} = \frac{1}{C^2} \quad (A > B > C)$$

的三个整数  $A, B, C$ , 求:

(1)  $A, B, C$  之和小于 100 的倒勾股数有多少组? 答案: 2

(2) 在(1)中  $A, B, C$  之和最小的是哪组? 答案: 20, 15, 12

13. 满足下列两个条件:

(1) 千位数字与百位数字相同(非 0), 十位数字与个位数字相同;

(2) 是某两位数的平方。

的四位正整数称为四位平方数。例如:

$$7744 = 88^2$$

所以 7744 为四位平方数。求:

(1) 所有四位平方数的数目; 答案: 1

(2) 所有四位平方数之和。 答案: 7744

14. 用迭代法求  $y = \sqrt[3]{x}$  的值。 $x$  由键盘读入。利用下列迭代公式计算：

$$y_{n+1} = \frac{2}{3}y_n + \frac{x}{3y_n^2}$$

初始值  $y_0 = x$ , 误差要求  $\varepsilon = 10^{-4}$ 。

15. 用牛顿迭代法求方程  $e^{-x} - x - 0$  在  $x = -2$  附近的一个实根, 直到满足  $|x_{n+1} - x_n| \leq 10^{-6}$  为止。

(1) 求方程的根;

答案: 0.5671433

(2) 求当迭代初值为  $-2$  时的迭代次数。

答案: 6

16. 已知  $f(t) = \sqrt{\cos t + 4\sin(2t) + 5}$ , 求  $s = \int_0^{2\pi} f(t) dt$

(1) 将积分区间 100 等分, 利用矩形法求  $S$ 。

答案: 13.261250

(2) 将积分区间 100 等分, 利用梯形法求  $S$ 。

答案: 13.261250

17. 已知  $g(x) = \frac{f(f(x)+1)}{f(x)+f(2x)}$

$$\text{其中 } f(t) = \begin{cases} \frac{t}{1+\frac{t}{2}}, & 1 \leq t \leq 10 \\ 2t^2 + 3t - 5, & \text{其他} \end{cases}$$

求: (1)  $g(2.5)$ ;

答案: 4.043919E-01

(2)  $g(17.5)$ 。

答案: 272.841100

18. 一个自然数是素数, 且它的数字位置经过任意对换后仍为素数, 则称为绝对素数, 例如 13。试求所有二位绝对素数。 答案: 11, 13, 17, 31, 37, 71, 73, 79, 97

19. 求方程  $3x - 7y = 1$ , 在  $|x| \leq 100, |y| \leq 50$  内的整数解。

(1) 共有多少组整数解?

答案: 29

(2) 在上述各组解中,  $|x| + |y|$  最大值是多少?

答案: 143

(3) 在上述各组解中,  $x + y$  最大值是多少?

答案: 137

20. 求满足以下条件的  $x, y, z$ 。

(1)  $x^2 + y^2 + z^2 = 51^2$ ;

(2)  $x + y + z$  之值最大;

(3)  $x$  最小。

答案:  $x=22, y=31, z=34$

## 第四章 程序测试与调试

程序设计是一种创造性的活动,尤其是大型程序的设计要求由多人,经过需求分析、总体设计、详细设计、编码等一系列的活动才能完成。在这些活动中,人们犯错误的机会多得举不胜举,所以程序测试是保证程序质量的关键步骤。通过测试发现程序中的错误,然后找出错误的原因和位置并加以改正,这就是程序调试的目的。本章介绍程序测试与调试的一般方法。

### 4.1 程序测试

测试就是用精心设计的数据去运行程序,从而发现程序中的错误。一般地说,程序测试的基本方法有两种:黑盒法与白盒法。

黑盒测试(Black box testing)也称为功能测试。它只着眼于程序的外部特性,即程序能满足哪些功能。测试程序能否正确接收输入,并能否得到正确的输出结果。这种测试不考虑程序的内部逻辑结构。衡量测试数据设计的好坏,是看它们能测试程序的哪些功能。

白盒测试(White box testing)也称为结构测试。它着眼于程序的内部结构,对程序的各个逻辑路径进行测试,在不同点上检查程序的状态,看它的实际状态与预期的状态是否一致。测试数据设计的好坏,在于它能够覆盖程序逻辑路径的程度。

一般说来,在制定测试方案时,若程序要实现的功能是已知的,就可以采用黑盒测试法,若程序内部的逻辑结构已知,就可以采用白盒测试法。但是,这两种测试法都不可能进行完全测试。正如 Dijkstra 所说,“测试只能说明程序有错,不能证明程序无错”,对黑盒测试来说,要进行完全测试,必须使用穷举输入测试,即把所有可能的输入数据都作为测试数据全部测试一次,才能得到完全测试,但这是难以实现的。

使用白盒法选择测试数据,为了做到完全测试,要把程序中每条可能的通路都执行一次,即使对很小的程序,通常也做不到这一点。由于程序完全测试的不可能性,所以测试阶段要考虑的基本问题是:使用有限个测试数据,尽可能多发现一些错误。

测试阶段应该注意的一些基本原则是:

(1) 测试用例应该由输入数据和预期的输出结果两部分组成。这就是说,在执行程序之前应该对期望的输出有明确的描述,这样,测试后可将程序的输出同它仔细地对照检查。若不事先确定预期的输出,就可能把似乎是正确而实际是错误的结果当成是正确的

结果。

(2) 测试用例不仅要选择合理的输入数据,还应选用不合理的输入数据。许多人往往只注意前者而忽略了后一种情况,为了提高程序的可靠性,输入数据不合理的各种情况是应该认真检查的。

(3) 除了检查程序是否做了应做的工作之外,还应检查程序是否做了不应做的事。

(4) 应该长期保留所有的测试用例,以便再测试时使用。

测试阶段又分如下的步骤:

(1) 模块测试。又称单元测试,检查每个模块是否有错误,主要发现详细设计阶段的错误。

(2) 组装测试。又称综合测试,要检查模块之间的接口的正确性,主要用于发现概要设计阶段的错误。

(3) 确认测试。要检查程序系统是否满足用户的功能要求,主要用于发现需求分析阶段的错误。

## 4.2 程序调试

程序经过测试暴露许多错误,还必须进一步诊断错误的原因和位置,进而改正程序中的错误,这就是程序调试的任务。也就是说,调试过程由两个步骤组成,首先根据错误的迹象,确定错误的准确位置,然后仔细研究这段程序以确定问题的原因并设法改正错误。其中第一步所需要的工作量大约占调试工作总量的 95%,因此,程序调试重点是如何确定程序错误的位置。

有人把问题的外部现象称为错误,把问题的内在原因称为故障。使用这个术语,在测试暴露一个错误之后,进行调试以确定与之相联系的故障。一旦确定了故障的位置,则修改源程序以便排除这个故障。为了确定故障,可能需要进行某些诊断测试,在排除故障之后,为了保证故障确实被排除了,错误确实消失了,需要重复进行暴露这个错误的原始测试以及某些回归测试。如果所做的改正是无效的,则重复上述过程直到找到一个有效的解决办法。

程序调试是程序设计过程中一项艰巨的脑力劳动。调试开始时,仅仅面对着错误的征兆,然而在问题的外部现象和内在原因之间并没有明显的联系。在组成源程序的数以万计的元素(语句、数据结构等)中,每一个都可能是错误的根源。如何在浩如烟海的程序元素中找出有错误的那个元素,这是调试过程中最关键的技术问题。

### 4.2.1 诊断错误的实验方法

程序中的错误有时是十分隐蔽的。为了找到这些错误,首先要捕获一些与错误有关的线索,然后确定错误的原因和位置。下面介绍几种常用的错误诊断的实验方法。这些方法也常常联合起来使用。

### 1. 利用系统信息

源程序上机运行,语言处理系统和操作系统会在程序有错时给出信息。可以根据这些信息来找到程序中的错误并改正它们,必要时可查阅“错误信息表”。

#### (1) 编译过程中的错误

对解释型的高级语言来说,程序出现语法错误,系统便使程序在出错点中断,并指出错误的类型和位置。对编译型高级语言来说,编译系统把源程序检查一遍后,对语法错误会打印出一系列的错误信息,用户可以根据这些错误信息来分析错误性质和原因,并进行相应修改。要注意的是,有时源程序中一个含糊的错误会引起编译程序的连锁反应,产生许多出错误信息,在这种情况下,往往只需纠正一个出错信息对应的地方即可。例如,若程序中数组说明语句有错,这时,那些与该数组有关的程序行都会被编译系统检查出错。在这种情况下,只要修改了数组说明语句的错误,其余错误就会同时消失。

#### (2) 连接过程中的错误

在连接过程中要涉及到模块之间以及模块与系统之间的关系。如果程序中有外部调用、存储区设置和各模块之间的接口等方面的错误,连接程序就会提示出错的信息。

#### (3) 运行过程中的错误

虽然经过编译和连接过程中的调试已排除了程序中的一些错误,但并不能说程序就一定能正确运行了。恰恰相反,程序运行时往往还有错误,需要继续调试。

程序运行中的错误大体上可以归纳为两类,一类是运行程序时计算机系统给出错误信息,这类错误比较多的是与数据的输入输出格式有关、与文件操作有关;另一类错误则表现为运行时系统不正常或结果不正确。如程序不能正常结束、没有任何输出结果或输出结果与预期的不一致等。

### 2. 插入调试语句

除了利用系统给出的信息进行分析、判断之外,还要采取一些调试手段。常用的方法是在程序中插入一些调试语句。

常用的调试语句有以下几种。

#### (1) 设置状态变量

在每个模块中设置一个状态变量,程序进入该模块时,便给该状态变量一个特殊值,根据各状态变量的值,可以判定程序活动的大致路径。

#### (2) 设置计数器

在每个模块或基本结构中,设置一个计数器,程序每进入该结构一次,便计数一次。这样,不仅可以判断程序的路径,而且当程序中有死循环时,用这种方法便能很快确定。

#### (3) 插入打印语句

打印语句是最常用的一种调试语句。打印语句用起来非常方便,能产生许多有用的信息。

打印语句有以下几种用法:

① “回声”打印。“回声”打印的特点是“读了就打”。它把打印语句放在紧靠读语句(或输入语句)之后、模块入口处或调用语句前后。可以帮助检查数据有没有被正确地输入或接口处信息传递是否正确。

② 追踪打印。追踪打印是为提供程序执行的路径信息而设置的打印语句。这些打印语句通常设置在模块首部或尾部、调用语句前后、循环结构内的第一个语句或最后一个语句、紧靠循环结构的后面第一个语句、分支点之前或分支中的第一个语句。

③ 抽点打印。抽点打印就是选择一些可疑点设置打印语句,以便打印有关变量的值。

### 3. 强制技术

强制技术就是利用调试用例,迫使程序逐个通过所有可能出现的执行路径,系统地排除无错的程序分支,逐步缩小检查的范围。

### 4. 借助调试工具

可以利用高级语言处理系统和操作系统提供的调试工具,进行单步、追踪运行。

#### 4.2.2 诊断错误的推理技术

程序调试还没有一套严密的理论,还无法只靠机械的、固定的方法去完成它,更多的是需要智慧和经验。

##### 1. 归纳法排错

归纳法排错是从特殊推及一般的排错方法,即首先汇集线索(所使用的测试用例及其结果、错误征兆等),经过整理,找出那些测试数据在什么地方、什么时候、能引起什么现象,然后对这些结论进行分析、归纳,提出错误原因及位置假想,最后再设计新的测试用例去验证这些假想。若能验证,便可以对照源程序找出错误原因,并对错误定位;若无法验证,说明假想错误或有多种错误,应提出新的假想,重新验证。

##### 2. 演绎法排错

演绎法排错是枚举所有可能引起出错的原因作为假设,然后逐一排除不可能发生的原因与假设,将余下的原因作为主攻方向。

应当指出的是,错误诊断的实验方法与推理技术应结合使用,互相补充。推理是在取得一定的实验数据的基础上进行的;推理得出的假设,要靠实验证明,并取得新的数据,把搜索范围缩小。

#### 4.2.3 错误修改的原则

这些原则在是显而易见的,但经常会遗忘和忽略。

(1) 要勤于思考。程序调试是分析问题、解决问题的过程,培养调试程序的能力,最有效的方法是勤于思考、积极分析、不断总结。

(2) 如果陷于困境,就把问题放到第二天去解决。若在适当的时间内(小程序半小时,大程序几个小时)找不到错误,则不要再考虑下去,隔一段时间,可能灵机一动就解决了这个问题。

(3) 陷于绝境以后,要与别人交谈自己的问题。可能在交谈的过程中,突然找到问题所在,或许别人的提示对自己有很大的启发。

(4) 避免用试验法。不要在问题没有搞清楚之前,就改动程序,这样对找出错误不利,程序越改越乱,以致于面目全非。

## 4.3 程序调试实例

### 4.3.1 程序的动态调试

一个 FORTRAN 程序编辑完成后,应先进行编译、连接生成可执行文件,然后运行。可以根据编译、连接和运行过程中的屏幕信息或运行结果来判断程序是否正确。通过 Fortran PowerStation 可以实现高效可视地编译和连接程序。

#### 1. 编译过程的调试

Fortran PowerStation 编译工具将源代码转变为一个或多个包含机器指令的目标文件。在源程序进行编译的过程中,对语法进行检查,如果发现源程序中有语法错误,就会显示错误信息,指出错误的类型和发现错误的行号。

一般有 3 种常见的编译错误:

(1) 一般性错误。例如,在块 IF 结构中,如果掉了 END IF 语句,编译器将会给出如下错误信息:

error FOR3592: missing END IF statement

出现这种错误时,编译器会对源程序继续编译。

(2) 严重警告。例如,在源程序的末尾处掉了 END 语句,编译器将会给出如下错误信息:

error FOR1768: missing END statement or missing program unit (possibly free-form source in fixed-form file) detected at <end of file>

(3) 警告。严格讲,警告并不是错误。但有时警告会提示给程序员一些可能的错误。例如,在调用子程序时,实参和虚参的类型不一致。主程序中实参为整型变量,而子程序 MYSUB 中虚参为实型变量,将会出现以下警告信息:

warning FOR4227: argument Y (number 2) in reference to procedure MYSUB from procedure main incorrect: has the wrong data type

可以根据这些错误信息找到程序中的错误并改正它们。为更清楚地了解程序的动态调试过程,下面以一个具体的排序程序为例介绍调试的过程,该程序所需读入的数据从 TEST.TXT 中读入,排序的结果放在 A 数组中。

**【例 4.1】** 输入的程序如下(保存为 arrange.f90):

```
PROGRAM ARRANGE
INTEGER, DIMENSION(:), ALLOCATABLE :: A
INTEGER TEMP
REWIND(UNIT=11)
OPEN (11, FILE='E:\TEST.TXT', FORM='FORMATTED', &
      STATUS='OLD')
```



```

READ(11,*)N
ALLOCATE(A(N))
DO K=1,N
    READ(11,*)A(K)
END DO
PRINT *, "THE DATA OF INPUT"
WRITE(*,*)(A(K),K=1,N)
DO I=1,N-1
    DO J=I+1,N
        IF(A(I).LT.A(J))THEN
            TEMP=A(I)
            A(J)=TEMP
            A(I)=A(J)
        ENDIF
    ENDDO
ENDDO
PRINT *, "THE DATA OF OUTPUT"
WRITE(*,*)(A(K),K=1,N)
CLOSE(11)
END

```

在主菜单中选择 Build 项,在下拉菜单中选择 Build arrange.exe 就可进行编译连接,在输出工作区的窗口就会列出各种错误警告信息,如图 4.1 所示。



图 4.1 编译错误信息

对 arrange.f90 编译共发现四处错误、一处警告。一般情况下,调试编译错误的最好办法是采用逐条修改的方式进行调试,即改正一条错误重新编译一次,直至程序完全无错

误。因为有时源程序中一个错误会引起编译程序的连锁反应,产生许多错误信息。

检查第 12 行,发现 WRITE 语句拼写错误引起出错信息。改正后再次编译,原来第 1,2 项错误提示消失。再检查第 15 行,发现丢失一个括号。改正后再编译,提示:

```
arrange.obj - 0 error(s), 0 warning(s)
```

这时,第一步调试完成。

要注意的是,有时候编译程序并没有给出错误行号或者给出的错误行号并不一定就是错误所在行,这时要前后分析,特别是对于一些固定的程序结构要更加注意。如下面的例子。

**【例 4.2】** 输入的程序如下(保存为 lpp1.f90):

```
S=0.0  
DO I=1,100  
S=S+1  
WRITE (*,*)S  
END
```

程序编译后错误信息如下:

```
E:\My Documents\lpp1.f90(5): error FOR3596: missing END DO statement
```

对于这样的问题,错误的实际位置就需要根据程序来分析判断。显然 DO 语句没有相对应的 END DO 语句。编译程序给出第 5 行有错,意思是程序结束了还没有对应的 END DO 语句。

## 2. 连接过程的调试

当项目编译器没有发现警告或错误信息后,项目连接器会马上对项目进行连接,连接没有发现错误,就会生成可执行文件,在结果输出工作区窗口中给出连接正确信息。如图 4.2 所示。

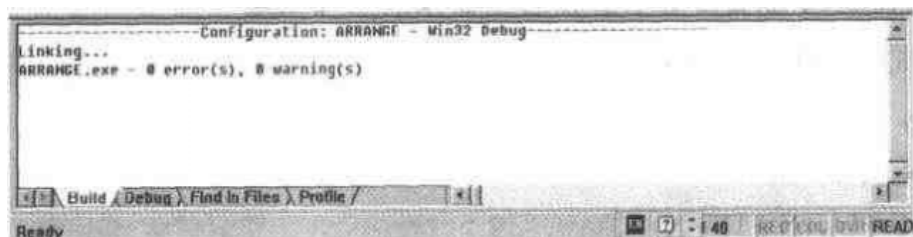


图 4.2 连接信息

如果程序中有外部调用、存储区设置和各模块之间的接口等方面的错误,连接程序就会提示出错的信息。通常一个比较容易出现的连接错误是语句函数或子程序没有定义或调用时名字输入错误。如下例的例子。

**【例 4.3】** 输入程序:

```
F1(N)=N*N+3*N+1  
S=0.0  
DO I=1,100  
S=S+F1(I)
```

```
END DO
WRITE (*,*)S
END
```

编译时没有出错,但连接时出现错误提示:

error LNK2001: unresolved external symbol \_F@4

fatal error LNK1120: 1 unresolved externals

原因是调用语句函数时函数名输入错误(F1 误输成 FI)。

### 3. 运行过程的调试

程序运行中的错误大体可以归纳为两类,一类在运行时计算机系统给出错误信息,另一类错误则表现为运行时系统不正常或结果不正确。对于一些简单的运行错误,可以通过对错误信息或程序的运行结果进行分析,找出问题的所在指出,并加以改正。

【例 4.4】 输入程序:

```
S=0.0
DO I=2,100
  S=S+1/I
END DO
P=(1+S*S)/S
WRITE (*,*)S
END
```

该程序编译连接都没有问题,但运行时给出下列信息:

Floating point division by zero

原因在于循环体内表达式的类型有问题。采用整型数据相除,结果为整数。这类错误可以通过在循环体内输出中间结果来发现。再看一个例子。

【例 4.5】 计算  $S = \sum_{n=1}^{10} \frac{1}{n(n+1)}$  的程序如下:

```
S=0.0
DO N=1,10
  S=S+1/((N+1)*N)
END DO
WRITE(*,*)S
END
```

该程序编译、连接均顺利通过,但运行结果为 0,显然不正确。错误原因与上例相同。

### 4.3.2 常用的调试语句

有时候不能直接判断出错误所在之处,常常使用调试语句。最常用的方法是插入打印语句和暂停语句,暂停语句(PAUSE 语句),一般与打印语句配合使用,便于检查程序执行情况。

【例 4.6】 从键盘输入 20 个数,按输入时的逆序打印出 20 个数。要求输入时每行

输入 5 个数,输出时每行打印 10 个数。

输入程序如下:

```
DIMENSION X(20)
READ(*,10)X
DO I=1,20
  X(I)=X(21-I)
END DO
WRITE(*,20)(X(I),I=1,20)
10 FORMAT(5F4.1)
20 FORMAT(1X,10F5.1)
END
```

输入:

```
1.0  2.0  3.0  4.0  5.0
6.0  7.0  8.0  9.0  10.0
11.0 12.0 13.0 14.0 15.0
16.0 17.0 18.0 19.0 20.0
```

输出:

```
20.0 19.0 18.0 17.0 16.0 15.0 14.0 13.0 12.0 11.0
11.0 12.0 13.0 14.0 15.0 16.0 17.0 18.0 19.0 20.0
```

显然没有达到要求。为了找出错误,在 READ 语句后插入 WRITE(\*,20)(X(I), I=1,20)。运行后发现 WRITE(\*,20)(X(I),I=1,20)的输出与输入数据一致,说明是后面的循环结构有错。考察输出结果,发现前 10 个数据还是正确的,从第 11 个数据开始出现错误。原来当  $I=11$  时,  $X(I)$  已经不是原来的  $X(I)$  了。修改程序如下:

```
DIMENSION X(20),Y(20)
READ(*,10)X
DO I=1,20
  Y(I)=X(21-I)
END DO
WRITE(*,20)(Y(I),I=1,20)
10 FORMAT(5F4.1)
20 FORMAT(1X,10F5.1)
END
```

设置计数器也是常用的调试方法,如下面的例子。

**【例 4.7】** 用牛顿迭代法求  $f(x)=x^3-2x^2+4x+1=0$  在  $x=0$  附近的实根。

输入程序如下:

```
READ(*,*)X1
DO WHILE(ABS(X-X1).GT.1E-16)
  F=X1**3-2.0*X1**2+4.0*X1+1.0
```

```

F1=3.0 * X1 * * 2-4.0 * X1-4.0
X=X1-F/F1
X1=X
END DO

```

```
PRINT *, 'X1=', X1, 'X=', X
```

```
END
```

输入 1.5, 运行结果为:

```
X1= 2.631579E-01 X= 2.631579E-01
```

验根后发现 X 并不是方程的根。这时为了确认问题出在那里, 可在程序中引入计数器 N, 在 READ 语句后插入 N=0, 在循环体中插入语句 N=N+1 并在循环体 END DO 前插入打印语句:

```
PRINT *, 'N=', N, 'X1=', X1, 'X=', X
```

运行后输出 N=1, 说明只执行了一次循环体就结束了, 而且这时 X1 和 X 的值相等。分析问题, 发现问题出在 X1=X, 此处是为了给下一次循环准备好初值, 但这使循环条件表达式为假, 循环退出。

程序修改如下:

```

READ(*,*) X
X1=1
N=0
DO WHILE(ABS(X-X1).GT.1E-16)
    X1=X
    F=X1**3-2.0 * X1**2-4.0 * X1+1.0
    F1=3.0 * X1**2-4.0 * X1+4.0
    X=X1-F/F1
    N=N+1
END DO
PRINT *, 'N=', N, 'X1=', X1, 'X=', X
END

```

输入 1.5, 输出结果为:

```
N= 6 X1= -2.224945E-01 X= -2.224945E-01
```

经过验算可知程序结果是正确的。

在程序调试中, 有时需要暂时删除某些语句。这时最好的办法是在语句前加!, 把该语句变为注释语句, 以免重新输入。

### 4.3.3 系统调试工具 Debugger

#### 1. 启动 Debugger

当程序能够运行但不能输出预期结果, 对于简单的错误可以使用前面的方法调试, 但有时需要知道程序执行的情况, 使用系统调试工具 Debugger 可以调试 FORTRAN 应用

程序、设置断点、查看程序或动态库的运行状态或在程序的任何一点测试程序、变量和寄存器状态。

启动 Debugger,可以在 Build 菜单中选择 Debug 选项,该选项又有三个子项可供选择:

- (1) Go 选项,此选项执行程序直到有断点时停止或直到程序的结束;
- (2) Step into 选项。选此项单步执行;
- (3) Run to cursor 选项。此选项执行程序到光标所在的位置。

当项目调试器启动后主菜单中的 Build 项变成了 Debug 项。在 Debug 中包含如下调试命令:

- (1) Go 命令,执行程序到有断点时停止或直到程序结束;
- (2) Restart 命令,重新运行程序;
- (3) Stop Debugging 命令,结束程序的调试;
- (4) Break 命令,在当前行中断程序的执行;
- (5) Step Into 命令,单步执行程序;
- (6) Step Over 命令,单步执行程序直到遇到子程序或函数调用。此时只执行调用,但不在调用的过程中单步执行;
- (7) Step Out 命令,程序执行到 RETURN 或 STOP 止;
- (8) Run to Cursor 命令,执行程序到光标所在位置。

## 2. 设置断点

所谓断点,是告诉项目调试器在什么时候暂停程序,以便对变量或有关内容进行检查。断点只能设在可执行语句行、子程序或函数的开始处。

设置断点可以在主菜单 Edit 中选择 Breakpoints 选项,或使用快捷键 F9 或单击工具条的增删断点键,如图 4.3 所示。



图 4.3 设置断点

为了表示断点的存在,项目调试器在程序行的开头显示一个红颜色的圆点。以例4.1程序为例,该程序编译连接都已调试通过,但运行时结果不正确,可以在程序行 DO J=I

+1,N 前设置断点,然后选择 Go 运行。

### 3. 查看和修改变量的值

在启动 Debugger 后,若选择 Go 命令,则项目自动执行到所设置的断点处暂停,同时自动启动项目调试器中的 Watch 工具和 Variables 工具。

在 Variables 窗口(图 4.4)中列出了程序中出现所有变量,在 Watch 窗口(图 4.5 和图 4.6)中可以输入要查看的变量名,立即在其右边的 Value 表中出现所要查看的变量的值,用鼠标点击 Value 表中对应栏可以改变变量的值。

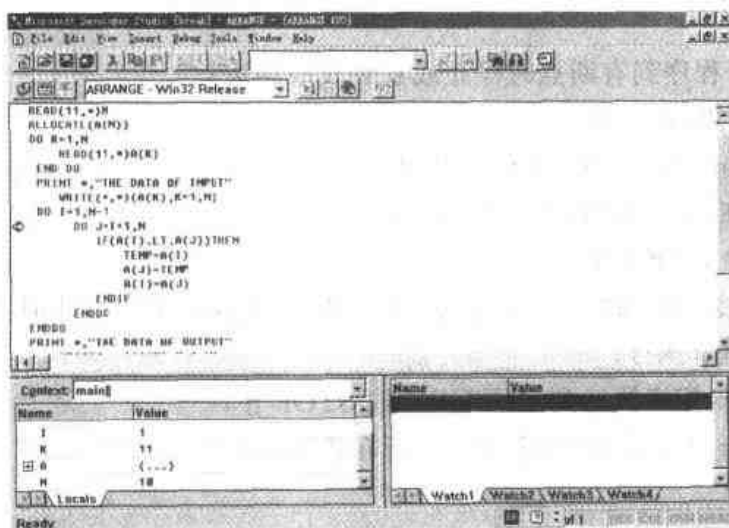


图 4.4 Variables 窗口

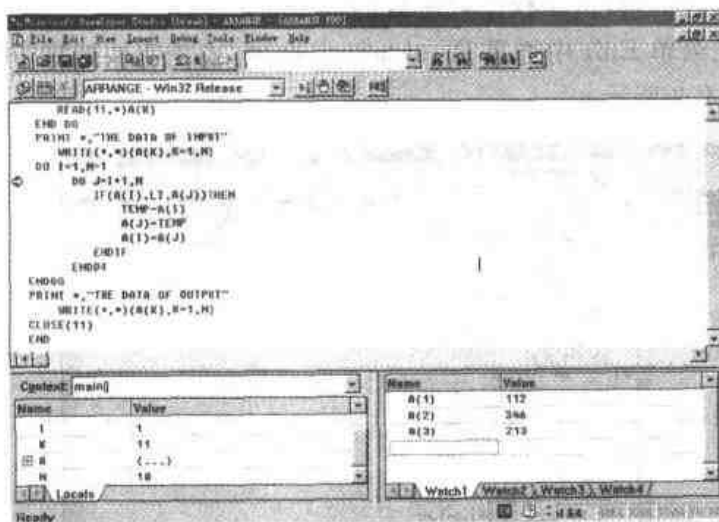


图 4.5 Watch 窗口

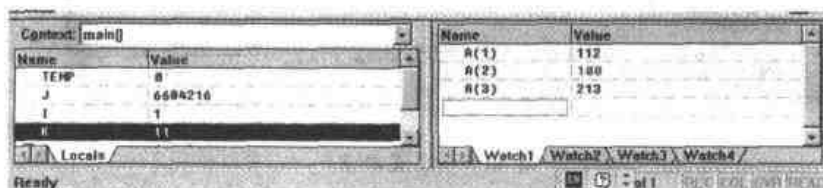


图 4.6 改变 A(2)的值后的 Watch 窗口

## 第五章 习题选解

### 5.1 填空题

1. 在 FORTRAN 90 中一行最多可以有\_\_\_\_\_个字符,若一条语句超过了最大允许长度,需要在语句末尾加\_\_\_\_\_。
2. 在对变量的类型说明方式中,优先级别最高的是\_\_\_\_\_。
3.  $\frac{2}{3}\sin\frac{\pi}{4}$  的 FORTRAN 表达式是\_\_\_\_\_。
4. 已知  $K=3, Y=1.5$ , 执行赋值语句  $X=1/K * K - K$  后  $X$  的是\_\_\_\_\_, 执行赋值语句  $I=1/K * Y + Y$  后  $I$  的值是\_\_\_\_\_。
5. 不等式  $1.0 < x < 3.0$  的 FORTRAN 表达式是\_\_\_\_\_。
6. 变量  $A$  和变量  $B$  同号的 FORTRAN 表达式是\_\_\_\_\_。
7.  $M$  不能够被  $N$  整除 FORTRAN 表达式是\_\_\_\_\_。
8. 已知  $K=2, J=3, A=.TRUE.$ , 则 FORTRAN 表达式  $K-J \leq K$ . AND(. NOT. A). OR. ( $K+J \geq J$ ) 的值是\_\_\_\_\_。
9. 已知  $A=3.5, B=5.0, C=2.5, D=1.0$ , 则 FORTRAN 表达式  $A \geq 0$ . AND.  $A + C$ . GT.  $B + D$ . OR. . NOT. . TRUE. 的值是\_\_\_\_\_。
10. 下面的格式输入语句  
READ(\*,100)X  
100 FORMAT(3F3.1)  
若输入 1234.675, 则  $X$  的值是\_\_\_\_\_。
11. 阅读下列程序:  
READ(\*,200)P,Q,X  
WRITE(\*,200)X  
200 FORMAT(1X,3(2X,F3.1))  
END  
执行这个程序时,若从键盘输入 1234567890123456789, 则输出为\_\_\_\_\_。
12. 循环语句 DO I=4,2,2 所确定的循环次数为\_\_\_\_\_。
13. 循环语句 DO I=2.5,-1.5,-0.5 所确定的循环次数为\_\_\_\_\_。



14. 下列双重循环中,循环体执行的总次数为\_\_\_\_\_。

```
DO T=1.5,10.5,1.5
```

```
  DO I=1,10
```

```
    <循环体>
```

```
  END DO
```

```
END DO
```

15. 下列双重循环中,循环体执行的总次数为\_\_\_\_\_。

```
DO J=1,10
```

```
  DO I=J,10
```

```
    <循环体>
```

```
  END DO
```

```
END DO
```

16. 下列双重循环中,循环体执行的总次数为\_\_\_\_\_。

```
DO J=1,5,2
```

```
  DO I=10,3
```

```
    <循环体>
```

```
  END DO
```

```
END DO
```

17. 设 C 是复型变量, A, B 是实型变量, 把 A 的值赋给 C 的实部, 把 B 的值赋给 C 的虚部的赋值语句是\_\_\_\_\_。

18. 子例行程序必须以\_\_\_\_\_语句开头, 以 END 语句结束。

19. 语句 INTEGER A(3,4) 定义了一个整型数组, A 数组的第 9 个元素是\_\_\_\_\_。

20. LEN(M(3:5)//N(1:4)//K(2:4)) 的值是\_\_\_\_\_。

21. 'AB'///'CD'///'EFG' 的值是\_\_\_\_\_。

22. 给不同的程序单位的变量分配同一存储单元应使用\_\_\_\_\_语句。给同一程序单位中的不同变量分配同一个存储单元应使用\_\_\_\_\_语句。

## 答 案

- |                  |     |                          |                                |
|------------------|-----|--------------------------|--------------------------------|
| 1. 132           | 续行符 | 2. 类型说明语句                | 3. 2.0/3.0 * SIN(3.1415927/4)  |
| 4. 3.0           | 1   | 5. 1.0 < X .AND. X < 3.0 | 6. SIGN(A,B) == A 或者 A * B > 0 |
| 7. MOD(M,N) /= 0 |     |                          | 8. .TRUE.                      |
| 9. .FALSE.       |     | 10. 12.3                 | 11. □□ * * *                   |
| 12. 0            |     |                          | 12. 0                          |
| 13. 9            |     | 14. 70                   | 15. 55                         |
|                  |     |                          | 16. 0                          |
| 17. C=CMPLX(A,B) |     | 18. SUBROUTINE           | 19. A(3,3)                     |
|                  |     |                          | 20. 10                         |
| 21. ABCDEFG      |     | 22. COMMON               | EQUIVALENCE                    |

## 5.2 阅读程序题

阅读下列程序,写出程序运行结果。

1. INTEGER::ID=1,K=1

REAL::SUM=1

DO WHILE(ID.LE.7)

K=-1.0\*K

ID=ID\*2

SUM=SUM+K/ID

END DO

PRINT \*,SUM

END

2. IMPLICIT INTEGER(X-Z)

Y1=5.0/2.0+3.8

PRINT \*,Y1

END

3. READ(\*,100) IA,I

IF(IA>125) THEN

I=I+10

ELSE

I=I-10

END IF

PRINT \*,I

100 FORMAT(I2,I1)

END

输入 186

4. DATA X,Y,Z/2\*-1.0,1.0/

A=300\*Z-0.1\*X-0.01\*Y

WRITE(\*,100) A

100 FORMAT(F7.3)

END

5. LOGICAL X,Y

READ(\*,100)X,Y

WRITE(\*,100)X,Y

100 FORMAT (1X,L4,2X,L3)

END

输入 FTTF

6. LOGICAL L1,L2,P,Q

INTEGER X,Y

DATA X,Y,L1,L2/2 \* -1,2 \* .FALSE. /

P=.NOT. (L1. AND. L2. OR. X>Y)

Q=.NOT. (L1. AND. L2. OR. X>=Y)

WRITE( \*, \* )P,Q

END

7. READ \*,X

IY=0.0

IF(X>10.0) IY=2 \* X+1.0

IF(X>5.0) IY=X+6.0

IF(X>0) IY=10.4

WRITE( \*, \* )IY

END

输入 12.5

8. DATA M,N/8,24/

L=MOD(M,N)

DO WHILE(L>0)

M=N

N=L

L=MOD(M,N)

END DO

PRINT \*,N

END

9. READ \*,X

IF(X<0.0)THEN

Y=0.0

ELSE IF((X<5.0). AND. (X/=2.0))THEN

Y=1.0/(X+2.0)

ELSE IF(X<10.0) THEN

Y=1.0/X

ELSE

Y=10.0

ENDIF

PRINT \*,Y

END

输入 2.0

10. LOGICAL P

READ \*,X,Y

P=, TRUE.

IF (X+Y<X) P=, FALSE.

Z=Y

IF(P) THEN

X=X+Z

ELSE

Z=X

END IF

WRITE(\*,\*) Z

END

输入 3.6 , -1.2

11. READ (\*,\*)X

IF(X<0) THEN

Y=0.0

ELSE IF (X>1.0) THEN

IF(X>=5.0) THEN

Y=5.0

END IF

ELSE

Y=0.5

END IF

PRINT \*,Y

END

输入 0.7

12. LOGICAL P

DATA M,N/5,7/

READ \*,IX

P=MOD(IX,M)==0. OR. MOD(IX,N)==0

IF(P) THEN

WRITE(\*,\*)'Y'

ELSE

WRITE(\*,\*)'X'

END IF

END

输入 28

```
13. DO K=5,1,-1
    M=0
    DO J=K,5
        M=M+K * J
    END DO
END DO
WRITE( *, * ) M
END

14. DO I=1,2
    S=0.0
    DO J=1,3
        S=S+I+J
    END DO
END DO
WRITE( *, * ) S
END

15. READ *,N
    F=1.0
    T=1.0
    K=1
    DO WHILE (K<=N)
        K=K+1
        F=-F
        T=T * K * F
    END DO
    PRINT *,T
END
```

输入 5

```
16. READ ( *, * )N
    S=0
    F=1
    DO K=1,N
        F=F * K
        S=S+F
    END DO
    WRITE( *, * )S
END
```

输入 5

```
17. DO I=1,10
    S=0.0
    DO J=I+1,10
        S=S+I+J
    END DO
END DO
WRITE(*,*) S
END
```

```
18. READ *,K
    SUM=1.0
    J=1
    DO WHILE(I<=K)
        F=1
        DO I=2,2*J+1
            F=F*I
        END DO
        SUM=SUM+F
        J=J+1
    END DO
    PRINT *,SUM
END
```

输入 3

```
19. IMPLICIT NONE
    INTEGER I,J,K
    INTEGER,DIMENSION(2)::N
    DO I=1,2
        N(I)=0
    END DO
    K=2
    DO I=1,K
        DO J=1,K
            N(J)=N(I)+1
        END DO
    END DO
    WRITE(*,100)N(K)
    100 FORMAT(1X,I1)
END
```

```
20. INTEGER A(3,3)
```

```
DATA A/1,2,3,4,5,6,7,8,9/  
WRITE(*,100)((A(K,J),J=1,K),K=1,3)  
100 FORMAT(1X,10I2)  
END
```

21. INTEGER A(3,4)

```
DATA A/4*0,4*1,4*2/  
WRITE(*,100)((A(I,J),I=1,3),J=1,4)  
100 FORMAT(1X,4I3)  
END
```

22. DIMENSION A(4,3)

```
DATA A/3*-1.0,3*-2.0,3*-3.0,3*-4.0/  
S=0.0  
DO I=1,4  
  D=ABS(A(1,1))  
  DO J=2,3  
    IF(ABS(A(1,J))>D) THEN  
      D=ABS(A(1,J))  
    END IF  
  END DO  
  S=S+D  
END DO  
WRITE(*,*)S  
END
```

23. DIMENSION M(3,3)

```
DATA M/1,2,3,4,5,6,7,8,9/  
DO I=1,3  
  WRITE(*,100)(M(I,J),J=1,3)  
END DO  
100 FORMAT(1X,3I2)  
END
```

24. DIMENSION X(3,3)

```
DATA X/1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0/  
F=1.0  
DO I=1,3  
  F=F*X(I,4-I)  
END DO  
WRITE(*,*)F  
END
```

```
25. INTEGER A(2,3),B(3,2),C(2,2)
DATA A/2*1,4*2/,B/3*3,3*1/
DO I=1,2
  DO J=1,2
    C(I,J)=0
    DO K=1,3
      C(I,J)=C(I,J)+A(I,K)*B(K,J)
    END DO
  END DO
END DO
WRITE(*,100)((C(I,J),J=1,2),I=1,2)
100 FORMAT(1X,2I3)
END

26. PARAMETER(N=10)
DIMENSION X(N)
DATA X/23.0,12.0,34.0,19.0,65.0,48.0,33.0,26.0,87.0,45.0/
P=X(1)
I=1
Q=X(1)
J=1
DO K=2,N
  IF (X(K)>P) THEN
    I=K
    P=X(K)
  END IF
  IF(X(K).LT.Q)THEN
    J=K
    Q=X(K)
  END IF
END DO
X(I)=X(N)
X(N)=P
X(J)=X(1)
X(1)=Q
S=X(1)+X(N)
WRITE(*,100) S
100 FORMAT(F4.1)
END
```



```
27. INTEGER A(3,3),B(3)
DATA A,B/9*0,3*1/
DO I=1,3
  DO J=1,3
    A(I,J)=I+J
  END DO
END DO
DO I=1,3
  DO J=I,3
    B(I)=B(I)*A(I,J)
  END DO
END DO
WRITE(*,100)(B(I),I=1,3)
100 FORMAT (1X,3I3)
END

28. INTEGER A(5),B(7),C(12)
DATA A/3,8,11,25,46/
DATA B/9,12,20,26,33,41,59/
K=1
I=1
J=1
DO WHILE(I<=5.AND.J<=7)
  IF(A(I)<B(J))THEN
    C(K)=A(I)
    K=K+1
    I=I+1
  ELSE
    C(K)=B(J)
    K=K+1
    J=J+1
  END IF
  IF(I>5)THEN
    DO L=K,12
      C(L)=B(J+L-K)
    END DO
  ELSEIF (J>7)THEN
    DO L=K,12
      C(L)=A(I+L-K)
```

```
END DO
ENDIF
END DO
IP=C(1)+C(12)
WRITE(*,*)IP
END
29. DIMENSION M(3,3)
DATA N,M/3,1,2,3,4,5,6,7,8,9/
S=0.0
DO K=1,N
    S=S+M(K,K)
END DO
S=S/N
PRINT *,S
END
30. INTEGER A(4,4)
DATA A/4*6,4*4,4*2,4*0/
X=F(A,2)
WRITE(*,100)X
100 FORMAT(1X,F5.2)
END
FUNCTION F(B,N)
INTEGER B(N,N)
F=1
DO I=1,N
    F=F*B(1,I)
END DO
END
31. X=1.0
Y=2.0
CALL SUB(X,Y,Z,3.0)
WRITE(*,100)X,Y,Z
100 FORMAT(1X,3F4.1)
END
SUBROUTINE SUB(A,B,X,Z)
A=A+1
B=B+2.0
Y=1.0
```

```
X=Y+Z
END
32. X=2.0
Y=3.0
CALL SS(X,Y)
WRITE(*,100)X,Y
100 FORMAT(1X,2F4.1)
END
SUBROUTINE SS(A,B)
A=A+A
W=A
A=B
B=W
END
33. INTEGER A(4,4),F
N=4
CALL SA(A,N)
WRITE(*,*)F(A,N)
END
SUBROUTINE SA(B,N)
INTEGER B(8,2)
DO N=1,2
DO M=1,8
B(M,N)=8*(N-1)+M
END DO
END DO
END
INTEGER FUNCTION F(B,N)
INTEGER B(N,N)
F=0
DO K=1,N
F=F+B(K,K)
END DO
END
34. REAL A(10),IP
DO K=1,10
A(K)=IP(K*2)/IP(K)
END DO
```

```
WRITE( *,100)A(1),A(2)
```

```
100 FORMAT(1X,2F4.1)
```

```
END
```

```
REAL FUNCTION IP(N)
```

```
M=0
```

```
DO K=1,N
```

```
    M=M+K
```

```
END DO
```

```
IP=M
```

```
END
```

35. EXTERNAL F

```
CALL S(F,3.0,4.0,W)
```

```
100 FORMAT(1X,F6.1)
```

```
WRITE( *,100)W
```

```
END
```

```
FUNCTION F(X,Y)
```

```
F=X*X+Y*Y
```

```
END
```

```
SUBROUTINE S(FUN,X,Y,Z)
```

```
Z=FUN(X,Y)
```

```
Z=FUN(Y,Z)
```

```
END
```

36. DIMENSION B(3,3)

```
DO K=1,3
```

```
    DO J=1,3
```

```
        B(K,J)=K+J
```

```
    END DO
```

```
END DO
```

```
Y=SUB(B,2,8)
```

```
WRITE( *,100)Y
```

```
100 FORMAT(1X,F5.1)
```

```
END
```

```
FUNCTION SUB(D,M,N)
```

```
DIMENSION D(N)
```

```
SUB=0.0
```

```
DO K=M,N
```

```
    SUB=SUB+D(K)
```

```
END DO
```

```
END
37. DIMENSION A(3,3)
DO K =1,3
DO J=1,3
A(K,J)=(K-1)*3+J
END DO
END DO
CALL S(A(2,1),3)
END
SUBROUTINE S(B,N)
DIMENSION B(N)
WRITE(*,100)(B(K),K=1,N)
100 FORMAT (1X,10F4.1)
END
38. CHARACTER*18 A,B,C*6,D*6,E*40
PARAMETER(C='*****',D='')
A=D//C//D
B=C//C//C
E='(1X,A18/1X,A18/1X,A18/1X,A18/1X,A18)'
WRITE(*,E)A,A,A,A,A
WRITE(*,E)B,B,B,B,B
WRITE(*,E)A,A,A,A,A
END
39. COMMON X,Y,Z,S
X=4.0
Y=5.0
CALL W
WRITE(*, '(1X,4F6.1)')X,Y,Z
END
SUBROUTINE W
COMMON E,F,G,H
E=E+F
G=E*F
H=E+G
END
```



### 5.3 填空编程题

1. 下面程序是求满足  $1^2 + 2^2 + 3^2 + \dots + K^2 < 2000$  的最大的  $K$  值。

```

IMPLICIT NONE
INTEGER::SUM=1,K=1
DO WHILE(     (1)     )
    K=K+1
    SUM=SUM+K*K
END DO
K=     (2)    
WRITE(*,*)K
END

```

2. 下列程序的功能是计算下列级数的值：

$$1 - \frac{1}{3} + \frac{1}{5} - \dots + \frac{(-1)^N}{2*N+1}$$

直到  $N$  的值等于 20 或者累加项小于  $10^{-3}$  为止。

```

REAL::SUM=1.0,D=1.0
INTEGER::N=0,F=1
DO WHILE(N<20     (1)     D>1.0E-3)
    N=N+1
    F=     (2)    
    D=1.0/(2*N+1.0)
    SUM=SUM+F*D
END DO
WRITE(*,*)SUM
END

```

3. 下列程序的功能是从键盘输入一个正整数,然后输出它所有的因子(不包括 1 和它本身)。

```

READ*,N
DO K=     (1)    
    IF(MOD(N,K)     (2)    ) THEN
        PRINT*,K
    END IF
END DO
END

```

4. 下列程序功能是从键盘依次输入一个班的各学生成绩,计算平均成绩,班上的最高成绩与最低成绩,程序中以输入一个负数作为结束。

```

N= (1)
SUM= (2)
READ *,X
AMAX=X
AMIN=X
DO WHILE ( (3) )
    N=N+1
    SUM=SUM+X
    IF(X>AMAX) AMAX=X
    IF( (4) ) AMIN=X
    READ *,X
END DO
AVE= (5)
WRITE(*,*)'AVE=',AVE
WRITE(*,*)'AMAX=',AMAX
WRITE(*,*)'AMIN=',AMIN
END

```

5. 下列程序的功能是:首先预置十名职工的年龄(以递增顺序排列),然后从键盘读入一个年龄,用对分法查找是否有此年龄的职工,有则输出“YES”,没有则输出“NO”。

```

INTEGER A(10)
(1)
DATA A/20,22,23,25,26,30,31,34,39,41/
READ(*,*)K
F= (2)
I=1
J=10
DO WHILE ((I<=J).AND.(.NOT.F))
    M=(I+J)/2
    IF(A(M)<K) THEN
        I= (3)
    ELSE IF(A(M)>K) THEN
        J= (4)
    ELSE
        F= (5)
    END IF
END DO
END DO

```



```

IF (F) THEN
  WRITE (*,*) 'YES'
ELSE
  WRITE (*,*) 'NO'
END IF
END

```

6. 下列程序功能是用迭代法求方程  $f(x) = x^3 + x - 1 = 0$  的根, 其中初值从键盘输入, 精度为 0.000001。

```

READ *, X0
X = X0 - 1
DO WHILE( (1) )
  (2)
  F = X * X * X + X - 1.0
  F1 = 3 * X * X + 1.0
  X0 = X - F/F1
END DO
PRINT *, X0
END

```

7. 下列程序的功能是, 从键盘输入一个整数 N, 然后计算并输出:

$$S = 1 + 2^1 + 2^2 + \cdots + 2^{|N|}$$

最后计算并输出:

$$T = 1 + \frac{1}{3} + \frac{1}{3} - \cdots + (-1)^{k+1} \frac{1}{K}$$

其中 K 为  $\sqrt{S}$  的整数部分。

```

READ(*,*) N
S = 0.0
(1)
DO I = 1, (2)
  S = S + F
  F = 2 * F
END DO
(3)
T = 0.0
F = 1.0
DO I = 1, K
  T = T + F/I
  F = (4)
END DO

```

```
WRITE(*,*)'T=',T
END
```

8. 下列程序功能是:从键盘输入一个正整数,如果该数为素数,则输出该素数,否则输出该数的所有因子(除去1与自身)。

```
LOGICAL P
READ(*,*)N
P=.TRUE.
K=2
DO WHILE( (1) )
  IF(MOD(N,K)==0)THEN
    WRITE(*,*) K
    (2)
  END IF
  K=K+1
END DO
IF( (3) )WRITE(*,*)N
END
```

9. 下列程序的功能是求2到500之间所有的亲密数对。亲密数对的定义为:如果M的因子(包括1,不包括本身)之和为N,而N的因子之和为M,则称M与N为一对亲密数。

```
DO M=2,500
  N=0
  DO K=1, (1)
    IF (MOD(M,K)==0) (2)
  END DO
  J=0
  DO K=1, (3)
    IF(MOD(N,K)==0) (4)
  END DO
  IF (( (5) ). AND. M<=N) THEN
    WRITE(*,*)M,N
  END IF
END DO
END
```

10. 下面程序的功能是:从键盘输入5组实数,每组有10个实数,并求出各组中元素绝对值之和的最大者和最小者。

```
REAL MAXX,MINN
DATA MAXX,MINN/0.0,0.0/
DO I=1,5
```

```

      (1)
      DO J=1,10
        READ *,X
        SUM=SUM+ (2)
      END DO
      IF(SUM>MAXX) (3)
      IF((I==1) (4) (SUM<MINN)) THEN
        MINN=SUM
      END IF
    END DO
    WRITE(*,*)'MAXX=',MAXX,'MINN=',MINN
  END

```

11. 下面程序的功能是从键盘输入一个班的成绩(不多于 50 人),然后计算各班的平均成绩,最后计算每一个学生与平均成绩的差。

```

      DIMENSION SCORE(50),D(50)
      DO I=1,50
        SCORE(I)=0
        D(I)=0
      END DO
      SUM=0.0
      N=0
      AVE=0.0
      (1)
      DO WHILE(X>0.0) THEN
        SUM=SUM+X
        (2)
        N=N+1
        READ(*,*)X
      END DO
      IF(N/=0) AVE=SUM/N
      DO I=1,N
        D(I)=SCORE(I)-AVE
      END DO
      WRITE(*,100)AVE
      100 FORMAT(1X,'AVE=',F10.2)
      WRITE(*,200)(I,D(I),I=1,N)
      200 FORMAT(1X,'D(',I2,',')=',F6.3)
    END

```

12. 下列程序功能是从键盘依次输入 10 个数据赋给一个长度为 10 的一维实型数组。在输入数据过程中,保持该数组中的元素是从小到大排列的,并且去掉值相同的数据,在数组最后剩余的元素均赋为 0,最后以格式编辑符 F10.5 输出数组中的元素,一行为 5 个。

```

DIMENSION A(10)
K=1
DO KK=1,10
  READ(*,*)X
  J=K-1
  DO WHILE(J>0.AND.X<A(J))
    (1)
    J=J-1
  END DO
  IF(X==A(J)) THEN
    DO I=J+1,K-1
      A(I)= (2)
    END DO
  ELSE
    A(J+1)= (3)
    K=K+1
  END IF
END DO
DO KK=K,10
  A(KK)= (4)
END DO
WRITE(*,100) (A(I),I=1,10)
100 FORMAT (1X, (5) )
END

```

13. 下面程序的功能是:在给定的二维数组中,从第 K 行起第 K 列后(不包括 K 行 K 列,K 从键盘输入)的元素中寻找绝对值最大的元素。

```

DIMENSION A(50,50)
DATA A/数据表略/
D= (1)
DO I=K,49
  DO J=K,49
    IF (( (2) ) > ABS(D)) D= (3)
  END DO
END DO
PRINT *, 'D=', D

```

END

14. 下列程序的功能是输出矩阵:

$$\begin{bmatrix} 3 & 5 & 7 & 9 \\ 7 & 9 & 11 & 13 \\ 11 & 13 & 15 & 17 \end{bmatrix}$$

的转置矩阵。

```
INTEGER A(3,4),B( (1) )
DATA A/3,7,11,5,9,13,7,11,15,9,13,17/
DO I=1, (2)
DO J=1, (3)
  B(I,J)= (4)
END DO
END DO
WRITE(*,100) ((B(I,J),J=1,3),I=1,4)
100 FORMAT(1X,3I4)
END
```

15. 下面程序是求数组 A 各行的最小值之和。

```
(1)
READ(*,100)A
100 FORMAT(I4)
S=0.0
DO K=1,4
  D=A(K,1)
  DO J=2,3
    IF( (2) ) D=A(K,J)
  END DO
  S=S+D
END DO
WRITE(*,*)S
END
```

16. 显示由 1,2,3,4 组成的全部 4 位数并统计这种数的个数。

```
N=0
DO I=1,4
  DO J=1,4
    DO K=1,4
      DO L=1,4
        M= (1)
        WRITE(*,*)M
```

```

        N= (2)
    END DO
END DO
END DO
END DO
WRITE( *, * )N
END

```

17. 下面程序的功能是:输入 100 个学生的考号和成绩,然后按成绩从高到低排序并输出。

```

DIMENSION N(100,2)
DO I=1,100
    READ( *, * )(N(I,J),J= (1) )
END DO
DO I=1,99
    K=I
    DO J= (2) ,100
        IF(N(K,2)<N(J,2)) (3)
    END DO
    IF(K/=I) THEN
        DO L=1,2
            M=N(I,L)
            (4)
            N(K,L)=M
        END DO
    END IF
    WRITE( *, * )N(I,1),N(I,2)
END DO
END

```

18. 下列程序的功能是:计算给定二维数组中每行元素的平均值,并依次存放在一个一维数组中。

```

PARAMETER(N=10)
DIMENSION A(N,N),B(N)
DATA A/数据表略/
DO I=1,N
    (1)
    DO J=1,N
        B(I)=B(I)+ (2)
    END DO
END DO

```

```

WRITE(*,100) (B(K),K=1,N)
100 FORMAT(1X,5F10.4)
END

```

19. 下列程序的功能是:从键盘读入一个班的学生学号(存放在一维数组 NUM 中)和四门功课的成绩(分别存放在二维数组 A 的第 1,2,3,4 列中),然后求出每人的平均分和总分(分别存放在二维数组 A 的第 5,6 列);最后按总分从高到低排出名次(存放在一维数组 IS 中)并输出每个学生的名次及成绩。

```

PARAMETER(N=50)
DIMENSION NUM(N),A(N,6),IS(N)
DO K=1,N
    READ(*,*)NUM(K),(A(K,J),J= (1) )
END DO
DO K=1,N
    (2)
    DO J=1,4
        A(K,6)=A(K,6)+ (3) )
    END DO
    A(K,5)= (4)
END DO
DO K=1,N-1
    DO J=K+1,N
        IF(A(K,6) (5) A(J,6)) THEN
            DO I=1,6
                P=A(K,I)
                (6)
                A(J,I)=P
            L=NUM(K)
                (7)
                NUM(J)=L
            END DO
        END IF
    END DO
    IS(K)= (8)
END DO
DO K=1,N-1
    IF(A(K,6) == A(K+1,6) ) (9)
END DO
DO K=1,N

```

```

WRITE( *,100)IS(K),NUM(K),(A(K,J),J=1,6)
END DO
100 FORMAT(1X,I2,2X,I7,2X,6F10.3)
END

```

20. 下列程序的功能是:在主程序中从键盘输入正整数  $M$  和  $N$  ( $M \leq N$ ), 然后求  $W = \sum_{K=M}^N K!$ , 其中  $K!$  通过调用子程序  $P$  来计算。

```

PROGRAM MAIN
READ( *, *)M,N
W= (1)
DO K= (2)
CALL P(Q, (3) )
W=W+ (4)
END DO
WRITE( *, *)'W=',W
END
SUBROUTINE P(S,N)
S= (5)
DO K=1,N
S=S (6)
ENDDO
END

```

21. 如果在  $m$  行  $n$  列矩阵中的元素  $A(I,J)$  既是第  $I$  行中的最小值, 又是第  $J$  列中的最大值, 则称  $A(I,J)$  为矩阵的鞍点。下列程序的功能是求出给定矩阵中的所有鞍点。

```

SUBROUTINE S(A,M,N)
INTEGER A(M,N),D,P
DO I=1,M
P= (1)
K=1
DO J=1,N
IF (A(I,J) (2) ) THEN
P=A(I,J)
K=J
END IF
END DO
DO J=K,N
IF(A(I,J) == P) THEN
D= (3)

```



```

      DO L=1,M
        IF(A(L,J) (4) D) THEN
          D=A(L,J)
        END IF
      END DO
      IF( (5) ) THEN
        WRITE(*,*)I,J,P
      END IF
    ENDIF
  END DO
END DO
END

```

22. 10000 元和 10000 元以上的货物征税 5%；5000 元到 10000 元的货物征税 4%；1000 元到 5000 元的货物征税 3%；1000 元以下的货物免税，输入货款求税金。

```

      READ *,X
      IF (X<1000)THEN
        TAX=0
      ELSE IF(X>=10000)THEN
        TAX= (1)
      ELSE IF(X>=5000)THEN
        TAX= (2)
      ELSE
        TAX=(X-1000)*0.03
      END IF
      PRINT *,X,TAX
    END

```

23. 下面程序的功能是计算给定二维数组各行的最大元素的平方和。

```

(1)
      READ(*,*)A
      WRITE(*,*)S(A,4,5)
    END
      FUNCTION S(B,M,N)
      DIMENSION B(M,N)
      S=0
      DO I=1,M
        (2)
        DO J=2,N
          IF(B(I,J)>D) D=B(I,J)

```

```

    END DO
    S=S+D*D
END DO
END

```

24. 该程序是找出一个二维数组中每行绝对值最大的元素所在的列。主程序与子程序采用无名公用区传递数据。

```

PROGRAM FALG
REAL B(20,20)
  (1)
  READ(*,'(20F6.2)')((B(I,J),J=1,20),I=1,20)
  DO K=1,20
    L=K
    (2)
    WRITE(*,300)L,N,D
  END DO
300 FORMAT(1X,I2,2X,I2,3X,F6.2)
END
SUBROUTINE LGV
REAL (3)
  (4)
  X=A(I,1)
  J=1
  DO K=2,20
    IF(ABS(A(I,K))>ABS(X))THEN
      A=A(I,K)
      (5)
    END IF
  END DO
END

```

25. 下列程序的功能是：将某文件打开，将数据分别赋给数组 M 和 Q，它们分别表示 5 个学生的学号和每人三门课程的分数。该程序用来按记录统计每个学生的学号、三门课程的分数。

```

PARAMETER(N=5,K=3)
DIMENSION Q(N,K),AVE(N)
CHARACTER *6 M(N)
OPEN(8,FILE='B95.DAT',STATUS= (1) )
DO I=1,N
  READ(8,100)M(I),(Q(I,J),J=1,K)

```

```

END DO
100 FORMAT(A6,4F6.1)
CLOSE(8)
CALL MEAN( (2) )
CALL PRI
CALL HEAD
CALL PRI
CALL TABLE(M,Q,AVE,N,K)
CALL PRI
END
! SUB-1
SUBROUTINE MEAN(S,A,N,K)
DIMENSION (3)
DO I=1,N
    SUM=0.0
    DO J=1,K
        SUM=SUM+S(I,J)
    END DO
    A(I)=SUM/K
END DO
END
! SUB-2
SUBROUTINE HEAD
    WRITE(*,100)
100 FORMAT(12X,'NO. M',5X,'Q1',5X,'Q2',5X,'Q3',5X,'AVE')
END
! SUB-3
SUBROUTINE TABLE(M,Q,A,N,K)
DIMENSION Q(N,K),A(N)
CHARACTER * 6 M(N)
DO I=1,N
    WRITE(*,100) (4)
END DO
100 FORMAT(11X,A,4F7.1)
END
! SUB-4
SUBROUTINE PRI
    WRITE(*,100)

```

```
100 FORMAT(10X,36(' - '))
END
```

## 答 案

1. (1) SUM < 2000      (2) K - 1
2. (1) .AND.      (2) -F
3. (1) 2, N/2      (2) == 0
4. (1) 0      (2) 0.0      (3) X >= 0.0      (4) X < AMIN      (5) SUM/N
5. (1) LOGICAL F      (2) .FALSE.      (3) M+1      (4) M-1      (5) .TRUE.
6. (1) ABS(X-X0) > 1E-6      (2) X=X0
7. (1) F=1.0      (2) ABS(N)      (3) K=SQRT(S)      (4) F=-F
8. (1) K <= N/2      (2) P=.FALSE.      (3) P
9. (1) M/2      (2) N=N+K      (3) N/2      (4) J=J+K      (5) M==J
10. (1) SUM=0.0      (2) ABS(X)      (3) MAXX=SUM      (4) .OR.
11. (1) READ(\*,\*)X      (2) SCORE(N+1)=X
12. (1) A(J+1)=A(J)      (2) A(1+1)      (3) X      (4) 0.0      (5) 5F10.5
13. (1) 0.0      (2) ABS(I+1, J+1)      (3) A(I+1, J+1)
14. (1) 4, 3      (2) 4      (3) 3      (4) A(J, I)
15. (1) INTEGER A(4, 3)      (2) A(K, J) < D
16. (1) I \* 1000 + J \* 100 + K \* 10 + L      (2) N+1
17. (1) 1, 2      (2) I+1      (3) K=J      (4) N(I, L) = N(K, L)
18. (1) B(I)=0.0      (2) A(I, J)/N
19. (1) 1, 4      (2) A(K, 6)=0.0      (3) A(K, J)      (4) A(K, 6)/4      (5) <  
     (6) A(K, I)=A(J, I)      (7) NUM(K)=NUM(J)      (8) K      (9) IS(K+1)=IS(K)
20. (1) 0.0      (2) M, N      (3) K      (4) Q      (5) 1.0      (6) \* K
21. (1) A(I, 1)      (2) <P      (3) A(I, J)      (4) >      (5) P==D
22. (1) (X-10000) \* 0.05 + 320      (2) (X-5000) \* 0.04 + 120
23. (1) DIMENSION A(4, 5)      (2) D=B(I, 1)
24. (1) COMMON B, L, N, D      (2) CALL LGV      (3) A(20, 20)  
     (4) COMMON A, I, J, X      (5) J=K
25. (1) 'OLD'      (2) Q, AVE, N, K      (3) S(N, K), A(N)  
     (4) M(I), (Q(I, J), J=1, K), A(I)

## 5.4 改错题

请改正程序中的错误,使其运行结果正确。注意:不得修改其他的程序行和整个程序的结构。

1. 输入若干个学生的成绩,分别统计小于 90 分,大于等于 90 分学生的人数,直到输入一个负数,程序结束。

```
INTEGER::GRADE,N1=0,N2=0
READ(*,*)GRADE
DO WHILE(GRADE>0)
    IF(GRADE<=90)THEN
        N1=N1+1
    ELSE
        N2=N2+1
    ENDIF
    READ(*,*)GRADE
ENDDO
WRITE(*,*)'成绩大于等于 90 分的人数为:',N1,'人'
WRITE(*,*)'成绩小于 90 分的人数为:',N2,'人'
END
```

2. 给定程序功能是计算  $1! + 2! + 3! + \cdots + M!$  的值。

```
COMMON N,W
READ(*,*)M
S=0
DO N=1,M
    CALL SS
    S=SS+W
END DO
WRITE(*,*)'S=',S
END
SUBROUTINE SS
COMMON N,W
W=1
DO K=1,N
    W=W*K
END DO
```

END

3. 给定程序功能是输入一个正整数,若该数是素数,则输出素数,否则输出该数的所有因子(除去1与该数本身)。

```
LOGICAL P
READ(*,*)N
P=.TRUE.
K=2
DO WHILE(K<=N/2)
  IF(MOD(N,K)==0)THEN
    WRITE(*,*)K
    P=.FALSE.
  ENDIF
  K=K+1
END DO
IF(.NOT.P)WRITE(*,*)N
END
```

4. 给定程序的功能是极坐标与直角坐标之间的转换。

```
READ(*,*)XR,YT,K
CALL CHAN(XR,YT,U,V,K)
IF(K==0)THEN
  WRITE(*,*)'X=',U,'Y=',V
ELSE
  WRITE(*,*)'R=',U,'T=',V
ENDIF
END
SUBROUTINE CHAN(A,B,C,D,KEY)
IF(KEY==0)THEN
  C=SQRT(A*A+B*B)
  D=ATAN(B/A)*180/3.1415926
ELSE
  C=A*COS(B*3.1415926/180.0)
  D=A*SIN(B*3.1415926/180.0)
ENDIF
END
```

5. 给定程序其功能是求本利和。公式为  $P1=P(1+R)^N$ ,  $P1$  表示  $N$  年后本利和,而  $P$  是存款数,  $R$  是利率,  $N$  是年数。

```
P=1000.0
R=0.05
```

```

N=3
P1=P*(1+R)*N
P1=INT(P1*100.0+0.5)/100.0
WRITE(*,100)P1
100 FORMAT(1X,'P1=',2X,F8.2)
END

```

6. 给定程序功能是判断整数  $N$  是奇数(ODD),还是偶数(EVEN),如是奇数打印, ODD,如是偶数则打印 EVEN。

```

READ(*,*)N
IF(MOD(N,2)==0)THEN
    WRITE(*,10)N
ELSE
    WRITE(*,20)N
ENDIF
10 FORMAT(1X,I5,'ODD')
20 FORMAT(1X,I5,'EVEN')
END

```

7. 给定程序其功能是建立一个存放学生成绩的顺序文件,将读入的数据存放到一个文件中,并输出到显示器上。课程门数为 3 门,由键盘输入学生人数  $N(N \leq 100)$ 。

```

PROGRAM EX01
DIMENSION NAME(100),S(100,3)
CHARACTER*12 NAME
WRITE(*,*)'INPUT NUMBER OF STUDENT'
READ(*,*)N
CLOSE(4,FILE='STUDENT.DAT',STATUS='NEW')
DO I=1,N
    READ(*,*)NAME(I)
    DO J=1,3
        READ(*,*)S(I,J)
    END DO
END DO
WRITE(4,200)(NAME(I),(S(I,J),J=1,3),I=1,N)
200 FORMAT(4X,A12,3F10.2)
WRITE(*,210)(NAME(I),(S(I,J),J=1,3),I=1,N)
210FORMAT(4X,A12,3F10.2)
CLOSE(4)
END

```

8. 给定程序功能是有有一个四位数  $F:DCBA$ ,头两位相同,后两位相同,此数又是某个

自然数  $N$  的平方,求此数。

```

INTEGER F,A,B,C,D
DO N=32,99
  F=N*N
  A=F-INT(F/10)
  B=INT(F/10)-INT(F/100)*10
  C=INT(F/100)-INT(F/1000)*10
  D=INT(F/1000)
  IF(A==B) THEN
    IF(C==D) THEN
      WRITE(*,120)F,N
    ENDIF
  ENDIF
END DO
120 FORMAT(1X,I4,' ',I2)
END

```

9. 给定程序功能是求 11 亿人口,如增长率分别为 1%, 1.5%, 2%, 2.5%, 3% 时,从 1995 年起,几年翻一番?

```

INTEGER Y
P=11E8
DO I=1,5
  R=(I+1)*0.005
  P1=P
  Y=1
  DO WHILE (P1<2*P)
    P1=P1*(1.0+R)
    Y=Y+1
  END DO
  WRITE(*,*)R,Y,P1
END DO
END

```

10. 给定程序其功能是打印金字塔图案如下:

```

      *
    * * *
  * * * * *

```

```

CHARACTER A,B*39
A='*'
DO I=1,3

```



```

      B=' '
      L=7-I
      DO J=1,2*I-1
        B(L+J:L+J)=A
      END DO
      FORMAT(*,*)B
    END DO
  END

```

11. 给定程序功能是输入  $8 \times 8$  的方阵 A 及 B, 使 A 的元素值为该元素的下标的和, 使 B 的元素值为该元素的下标的差。

```

      INTEGER A(8,8),B(8,8)
      DO I=1,8
        DO J=1,8
          A(I,J)=I+J
          B(I,J)=I-J
        END DO
      END DO
      WRITE(*,*)'A:'
      WRITE(*,100)((A(I,J),I=1,8),J=1,8)
      WRITE(*,*)'B:'
      WRITE(*,100)((B(I,J),J=1,8),I=1,8)
      100 FORMAT(1X,8I6)
    END

```

12. 给定程序功能是输入 4 个数 A,B,C,D, 按由大到小顺序把它们打印出来。请改正程序中的错误, 使其运行结果正确。

```

      READ(*,*)A,B,C,D
      IF(A<B)THEN
        T=A
        A=B
        B=T
      ENDIF
      IF(C<D)THEN
        T=C
        C=D
        D=T
      ENDIF
      IF(A<C)
        T=A

```

```

A=C
C=T
ENDIF
IF(B<C)THEN
  T=B
  B=C
  C=T
ENDIF
IF(D>B)THEN
  WRITE(*,*)A,D,B,C
ELSE
  IF(D>C) THEN
    WRITE(*,*)A,B,D,C
  ELSE
    WRITE(*,*)A,B,C,D
  ENDIF
ENDIF
END

```

13. 给定程序其功能是求  $e \approx 1 + 1/1! + 1/2! + 1/3! + \cdots + 1/N!$  的近似值。

```

REAL M
READ(*,*)N
E=0.0
M=1
DO I=1,N
  M=M/I
  E=E+M
END DO
WRITE(*,20)I,E
20 FORMAT(1X,I5,F15.6)
END

```

14. 给定程序功能是生成下面的图形。

```

* * * * *
* * * * *
* * * * *
* * * * *

```

```

CHARACTER LINE * 50
DO I=1,5
  LINE=""

```

```

N=25
DO L=1,5
    J=N+L+1
    LINE(J)='*'
END DO
WRITE(*,100)LINE
100 FORMAT(1X,A)
END DO
END

```

15. 给定程序功能是在方阵 A 中,使其对角线上的元素为 1,其余的元素为 0。

```

INTEGER A(6,6)
DATA A/36*1/
DO I=1,6
    A(I,I)=1
END DO
WRITE(*,100)((A(I,J),J=1,6),I=1,6)
100 FORMAT(4X,6I2)
END

```

16. 给定程序功能是键盘输入 10 个学生的学号和成绩存入一直接文件中。然后利用此文件,查找指定成绩的学生的学号,如找到,则打印出学生学号;如未找到,则打印出“NOT FOUND”。

```

OPEN(2,FILE='A.DAT',ACCESS='DIRECT',FORM='FORMATTED',&
    RECL=10)
DO I=1,10
    WRITE(2,REC=I)K,X
END DO
READ(*,*)Y
N=0
DO I=1,10
    READ(2,REC=I)K,X
    IF(X==Y)THEN
        N=N+1
        WRITE(*,*)K
    ENDIF
END DO
IF(N==0) WRITE(*,*)'NOT FOUND'
100 FORMAT(1X,I2,F7.2)
CLOSE(2)

```

END

17. 给定程序其功能是将任意三个实数 A, B, C 按从小到大的顺序输出。

```
REAL MAX, MIN
READ( *, * ) A, B, C
WRITE( *, * ) A, B, C
MAX = A
MIN = B
IF( A > B ) THEN
    MAX = B
    MIN = A
END IF
IF( C > MAX ) THEN
    WRITE( *, * ) MIN, MAX, C
ELSE IF( C > MIN ) THEN
    WRITE( *, * ) MIN, C, MAX
ELSE
    WRITE( *, * ) C, MIN, MAX
ENDIF
END
```

18. 给定程序其功能是输入二维数组, 变换后(元素被下标的积除)求所有元素之和。

```
DIMENSION A(3, 4)
READ( *, * ) M, N
S = 0.0
DO I = 1, M
    DO J = 1, N
        READ( *, * ) A(I, J)
        A(I, J) = A(I, J) / I * J
    END DO
END DO
DO I = 1, M
    DO J = 1, N
        S = S + A(I, J)
    END DO
END DO
WRITE( *, 500)
500 FORMAT(1X, 'S = ', F9.4)
END
```

19. 给定程序其功能是输入整数 A 和 B, 进行关系运算及算术运算。

```

INTEGER A,B
CHARACTER C*3,D*3
COMPLEX X,Y
READ(*,*)A,B
C='123'
D='456'
X=A<B.AND.A==0
Y=B<A.AND.B==0
IF(X)A=B
IF(Y)B=A
A=A+B
B=A+B
WRITE(*,10)A,C
WRITE(*,20)B,D
10 FORMAT(1X,2HA=,I4,3X,2HC=,A5)
20 FORMAT(1X,'B=',I4,3X,'D=',A5)
END

```

20. 给定程序其功能是有有一个不含形参的子程序 S, 求  $N!$  和  $1+2+3+\cdots+N$  的值。主程序中输入  $M$ , 通过调用  $S$  后, 计算出  $1 + \frac{2!}{1+2} + \frac{3!}{1+2+3} + \cdots + \frac{M!}{1+2+3+\cdots+M}$ 。主程序与子程序之间的数据传递是通过公用区来实现的。

```

SUBROUTINE S
COMMON N,W,V
W=1.0
V=0.0
DO I=1,N
    W=W*I
    V=V+I
END DO
END
COMMON N,W,V
READ(*,*)M
P=0
DO I=1,M
    N=I
    CALL S
    P=P+W

```

```

END DO
WRITE(*,*)'P=',P
END

```

21. 给定程序其功能是在主程序中输入  $K$  和  $M$  ( $K$  和  $M$  为正整数,  $K < M$ ), 通过调用子程序  $S$  求  $K! + (K+1)! + \cdots + M!$ , 子程序  $S$  的功能是求  $P = N!$ 。

```

SUBROUTINE S(P,N)
P=1
DO I=1,N
    P=P*I
END DO
END
READ(*,*)K,M
W=0.0
DO I=K,M
    CALL S(P,I)
    W=W+P
END DO
WRITE(*,*)'W=',W
END

```

22. 给定程序其功能是求两个矩阵的相乘积。

```

INTEGER A(2,3),B(3,2),C(2,2),T
READ(*,*)((A(I,J),J=1,3),I=1,2)
READ(*,*)((B(I,J),J=1,2),I=1,3)
DO K=1,2
    DO J=1,2
        T=1
        DO I=1,3
            T=T+A(K,I)*B(I,J)
        END DO
        C(K,J)=T
    END DO
END DO
WRITE(*,*)((C(I,J),J=1,2),I=1,2)
END

```

23. 给定程序其功能是用矩形法求  $x^2 + x + 1$  在  $[A, B]$  区间的定积分。

```

READ(*,*)N,A,B
X=A
H=(B-A)/N

```

```

F0=X * X+X+1.0
S=0.0
DO I=1,N
    SI=F0 * H
    S=SI
    X=X+H
    F0=X * X+X+1.0
END DO
WRITE(*,*)N,A,B,S
END

```

24. 给定程序其功能是输入 30 名学生一门课的考试成绩,以 0~9 分,10~19 分, ...,90~99 分,100 分为分数段,统计各分数段的人数。

```

DIMENSION M(11),S(30)
DATA M/11 * 0/
READ(*,*)(S(I),I=1,30)
DO I=1,30
    IF(S(I)<=100. AND. S(I)>=0)THEN
        K=S(I)/10
        M(K)=M(K)+1
    ENDIF
END DO
WRITE(*,*)(M(I),I=1,11)
END

```

25. 给定程序功能是插入一个任意的整数到 10 个有序的数中,插入后这些数仍然有序。主程序找到插入位置后,调用子程序 INSERT,将该数插入。

```

INTEGER A(11),X
READ(*,*)X
I=1
DO WHILE (I<=10. AND. X>=A(I))
    I=I+1
END DO
CALL INSERT (A,11,I,X)
WRITE(*,*)(A(K),K=1,11)
DATA A/2,4,8,11,15,20,23,32,40,45,0/
END
SUBROUTINE INSERT(B,N,J,X)
INTEGER B(N),X
DO K=N-1,J,-1

```

```

    B(K)=B(K+1)
  END DO
  B(J)=X
END

```

26. 给定程序,其功能是求数组 A(3,4)中值最小的元素 A(I,J)及其位置。

```

INTEGER A(3,4),P
READ(*,*)((A(I,J),J=1,4),I=1,3)
P=A(1,1)
M=1
N=1
DO I=1,3
  DO J=1,4
    IF(A(I,J)>P)THEN
      M=I
      N=J
      P=A(I,J)
    ENDIF
  END DO
END DO
WRITE(*,*)P,M,N
END

```

27. 给定程序,其功能是显示由 1,2,3,4 组成的全部四位数,并统计这种数的个数。

```

N=1
DO I=1,4
  DO J=1,4
    DO K=1,4
      DO L=1,4
        M=I*1000+J*100+K*10+L
        WRITE(*,*)M
        N=N+1
      END DO
    END DO
  END DO
END DO
WRITE(*,*)'N=',N
END

```

28. 给定程序,其功能是将数组 A(10)及 B(10)中对应的元素交换。

```

INTEGER A(10),B(10)

```



- |                            |                        |
|----------------------------|------------------------|
| 23. 错误: $S=SI$             | 正确: $S=S+SI$           |
| 24. 错误: $K=S(I)/10$        | 正确: $K=S(I)/10+1$      |
| 25. 错误: $B(K)=B(K+1)$      | 正确: $B(K+1)=B(K)$      |
| 26. 错误: $IF(A(I,J)>P)THEN$ | 正确: $IF(A(I,J)<P)THEN$ |
| 27. 错误: $N=1$              | 正确: $N=0$              |
| 28. 错误: $T=B(K)$           | 正确: $B(K)=T$           |
| 29. 错误: $D=B(I,J)$         | 正确: $D=B(1,1)$         |
| 30. 错误: $M=M+1$            | 正确: $M=M-1$            |

## 5.5 补充编程题

要求补充编制的内容放在! === FILLINGOVER === 的程序行上,其中已给出部分程序。

1. 编制从键盘输入 10 个数,用冒泡排序法按非递减顺序输出程序。

```

DIMENSION A(10)
READ(*,*)A
DO N=9,1,-1
    DO K=1,N
        ! === FILLINGOVER ===
    ENDDO
END DO
WRITE(*,*)A
END

```

2. 编程计算多项式  $1+1.1x+1.2x^2+1.3x^3+1.4x^4+1.5x^5$  当  $x=3.1$  的值。

```

PROGRAM PVALUE
DIMENSION A(6)
DATA A/1.0,1.1,1.2,1.3,1.4,1.5/,N,X/6,3.1/
Y=POLY(A,N,X)
WRITE(*,'(1X,A5,6F5.1)')A(K)=' ',A
WRITE(*,'(1X,A2,F5.1,A6,F8.2)')X=' ',X,';Y=' ',Y
END
! === FILLINGOVER ===

```

3. 编制程序用筛选法选出 N 以内的所有素数并显示。

```

PROGRAM PRIMES
PARAMETER(N=20)
INTEGER A(N),P(N)

```

```

M=M+1
X=1-sin(X0)
END DO
IF(M==0) THEN
  WRITE(*,*)'FAIL'
ELSE
  WRITE(*,*)'X=',X
ENDIF
END

```

## 答 案

- |   |                             |
|---|-----------------------------|
| 1. 错误: IF(GRADE<=90) THEN                         | 正确: IF(GRADE>=90) THEN      |
| 2. 错误: S=SS+W                                     | 正确: S=S+W                   |
| 3. 错误: IF(.NOT.P) WRITE(*,*) N                    | 正确: IF(P) WRITE(*,*) N      |
| 4. 错误: IF(K==0) THEN                              | 正确: IF(K==1) THEN           |
| 5. 错误: P1=P*(1+R)*N                               | 正确: P1=P*(1+R)**N           |
| 6. 错误: 10 FORMAT(1X,I5,'ODD')                     | 正确: 10 FORMAT(1X,I5,'EVEN') |
| 7. 错误: CLOSE(4,FILE='STUDENT.DAT','STATUS','NEW') |                             |
| 正确: OPEN(4,FILE='STUDENT.DAT',STATUS='NEW')       |                             |
| 8. 错误: A=F-INT(F/10)                              | 正确: 1; A=F-INT(F/10)*10     |
| 9. 错误: Y=1  | 正确: Y=0                     |
| 10. 错误: FORMAT(*,*) B                             | 正确: WRITE(*,*) B            |
| 11. 错误: WRITE(*,100)((A(I,J),I=1,8),J=1,8)        |                             |
| 正确: WRITE(*,100)((A(I,J),J=1,8),I=1,8)            |                             |
| 12. 错误: IF(A<C)                                   | 正确: IF(A<C) THEN            |
| 13. 错误: E=0.0                                     | 正确: E=1.0                   |
| 14. 错误: LINE(J)='*'                               | 正确: LINE(J:J)='*'           |
| 15. 错误: DATA A/36*1/                              | 正确: DATA A/36*0/            |
| 16. 错误: WRITE(2,REC=I) K,X                        | 正确: WRITE(2,100,REC=I) K,X  |
| 17. 错误: IF(A>B) THEN                              | 正确: IF(A<B) THEN            |
| 18. 错误: A(I,J)=A(I,J)/I*J                         | 正确: A(I,J)=A(I,J)/(1*J)     |
| 19. 错误: COMPLEX X,Y                               | 正确: LOGICAL X,Y             |
| 20. 错误: P=P+W                                     | 正确: P=P+W/V                 |
| 21. 错误: W=P                                       | 正确: W=W+P                   |
| 22. 错误: T=1                                       | 正确: T=0                     |

```

      DO I=1,10
        READ( *,* )A(I)
      END DO
      DO J=1,10
        READ( *,* )B(I)
      END DO
      DO K=1,10
        T=A(K)
        A(K)=B(K)
        T=B(K)
      END DO
      DO L=1,10
        WRITE( *,* )A(L),B(L)
      END DO
    END

```

29. 给定程序,其功能是计算并输出给定二维数组各行的最大元素的平方和。

```

      DIMENSION A(4,5)
      READ( *,* )A
      WRITE( *,* )S(A,4,5)
    END
    FUNCTION S(B,M,N)
      DIMENSION B(M,N)
      S=0
      DO I=1,M
        D=B(I,1)
        DO J=2,N
          IF(B(I,J)>D) D=B(I,J)
        END DO
        S=S+D*D
      END DO
    END
  END

```

30. 给定程序其功能是求一非线性方程  $x-1+\sin x=0$  在 1.0 附近的一个实根,精度要求为  $10^{-5}$ 。迭代格式为:  $x_{n+1}=1-\sin x_n$ 。如果迭代 10 次仍然不收敛,则终止运行。

```

      WRITE( *,* )'INPUT X,EPS,M'
      READ( *,* )X,EPS,M
      X0=0
      DO WHILE(ABS(X-X0).GE. EPS. AND. M>0)
        X0=X

```

```

DO K=1,N
  A(K)=K
END DO
! ===FILLINGOVER===
DO K=2,N
  IF(A(K)/=0)THEN
    P(M)=K
    M=M+1
  ENDIF
END DO
WRITE(*, '(1X,10I8)')(P(K),K=1,M-1)
END

```

4. 编程从存有  $N$  个字符的字符数组中删去指定的字符。

```

PARAMETER(NC=20)
CHARACTER C(NC),CH
INTEGER P,N
DATA CH/'C'/
WRITE(*,*)'Enter character''number'
READ(*,*)N
WRITE(*,*)'Enter string including',N,'char.'
READ(*,100)(C(I),I=1,N)
N0=N
P=1
! ===FILLINGOVER===
IF(N/=N0)THEN
  WRITE(*,200)(C(I),I=1,N)
ELSE
  WRITE(*,300)CH
ENDIF
100 FORMAT(20A)
200 FORMAT(1X,'String agter del.',1X,20A)
300 FORMAT(1X,'char',A,'could not be found!!')
END

```

5. 编程在一列有序数列中用折半查找法查找一个待查的数。

```

PARAMETER(N=15)
INTEGER A(N),TOP,BOT,MID,X
LOGICAL FIND
DATA A/10,17,29,22,31,44,51,68,73,81,92,124,135,146,158/

```

```

READ(*,*)X
FIND=.FALSE.
TOP=1
BOT=N
DO WHILE(TOP<=BOT, AND. .NOT. FIND)
    MID=(TOP+BOT)/2
    ! == FILLINGOVER ==
END DO
IF(.NOT. FIND) WRITE(*,100)X,'Has not found'
100 FORMAT(' /1X,2(I3,A))
END

```

6. 编制程序从一篇文章中查出其中包含有多少个“CHINA”，假使文章有 K 行，每行 80 个字符。

```

CHARACTER LINE * 80, WORD * 5
PARAMETER(WORD='CHINA')
READ(*,*)K
N=0
DO I=1,K
    READ(*,*)LINE
    J=1
    DO WHILE (J<=76)
        J=INDEX(LINE(J:80),WORD)
        ! - == FILLINGOVER ==
    END DO
END DO
WRITE(*,*)'THE NUMBER OF CHINA IS',N
END

```

7. 编制程序计算任意两个 20 位正整数之和精确值。

```

INTEGER A(21),B(21),C(21)
READ *,(A(I),I=2,21)
READ *,(B(I),I=2,21)
M=0
DO I=21,2,-1
    C(I)=A(I)+B(I)+M
    ! - - FILLINGOVER = - -
END DO
C(1)=M
WRITE(*,20)(C(I),I=1,21)

```

```
20 FORMAT(1X,'A=',21H)
```

```
END
```

8. 补充编制程序验证歌德巴赫猜想。一个不小于 6 的偶数必定能表示为两个素数之和。要求将 6—100 之间的全部偶数表示为两个素数之和。

```
INTEGER N,A,B,W
```

```
DO N=6,100,2
```

```
    DO A=3,N/2
```

```
        W=0
```

```
        J=SQRT(REAL(A))
```

```
        I=2
```

```
        ! ===FILLINGOVER===
```

```
        IF(W.EQ.0) THEN
```

```
            B=N-A
```

```
            W=0
```

```
            J=SQRT(REAL(B))
```

```
            I=2
```

```
            DO WHILE(1<=J. AND. W==0)
```

```
                IF(MOD(B,I)==0) THEN
```

```
                    W=1
```

```
                ELSE
```

```
                    I=I+1
```

```
                END IF
```

```
            END DO
```

```
            IF(W==0) WRITE(*,*)N,'=',A,'+',B
```

```
        END IF
```

```
    ENDDO
```

```
END DO
```

```
END
```

9. 编制程序把一个数组中的后  $N-1$  个元素依次向前移动一个位置, 第一个元素移动到最后面。

```
PARAMETER(N=10)
```

```
REAL A(N)
```

```
READ(*,*)A
```

```
T=A(1)
```

```
! ===FILLINGOVER===
```

```
PRINT *,A
```

```
END
```

10. 编制程序其功能是数组 P 中存放若干数, 现任意读入一个 X, 要求删除数组 P 中

与 X 相同的那些数(设 P 元素个数  $\leq 100$  且均为整数)。

```

INTEGER P(100),X
WRITE(*,*)'N=?'
READ(*,*)N
WRITE(*,*)'ENTER ARRAY P:'
READ(*,*)(P(I),I=1,N)
WRITE(*,*)'P=',(P(I),I=1,N)
WRITE(*,*)'ENTER X:'
READ(*,*)X
M=N
K=1
! === FILLINGOVER ===
IF(N/=M)THEN
    WRITE(*,*)'ARRAY P AFTER DEL:'
    WRITE(*,*)(P(I),I=1,N)
ELSE
    WRITE(*,100)X
ENDIF
100 FORMAT(1X,I6,'COULD NOT BE FOUND')
END

```

11. 编制程序,其功能是由键盘输入 N 与 M 的值,计算并输出  $SUM = (1+2+\cdots+N) + (1+2+\cdots+(N+1)) + \cdots + (1+2+\cdots+M)$  的值。

```

READ(*,*)N,M
SUM=0,0
DO K=N,M
    SUM=SUM+F(K)
END DO
WRITE(*,*)SUM
END
! === FILLINGOVER ===

```

12. 编制程序计算  $5+10+15+20+\cdots+65$ 。

```

K=0
I=5
! === FILLINGOVER ===
WRITE(*,20)K
20 FORMAT(1X,'K=',I5)
END

```

13. 补充编制程序计算  $0.2+0.4+0.6+\cdots+10.0$  及  $0.2^2+0.4^2+\cdots+10.0^2$ 。

```

S=0.0
T=0.0
X=0.2
! ===FILLINGOVER===
WRITE(*,10)S,T
10 FORMAT(1X,'S=',F10.4/1X,'T=',F10.4)
END

```

14. 补充编制程序求 10 个数  $X_1, X_2, \dots, X_{10}$  的  $R, R=X_{\max}-X_{\min}$ 。

```

DIMENSION X(10)
WRITE(*,*)'ENTER X'
READ(*,*)(X(I),I=1,10)
WRITE(*,100)(X(I),I=1,10)
XMAX=X(1)
XMIN=X(1)
C===FILLINGOVER===
R=XMAX-XMIN
WRITE(*,100)(X(I),I=1,10)
WRITE(*,*)'R=',R
100 FORMATE(1X,10F6.2)
END

```

15. 补充编制程序,其功能是输出 100 之内的 Fibonacci 数列各项。

```

I=1
J=0
! ===FILLINGOVER===
END

```

16. 补充编制程序其功能是给定  $N$  个元素的数组  $A$ ,在不增加其他数组的情况下,将数组  $A$  中元素的值按逆序重新进行存放(设  $N < 50$ )。

```

DIMENSION A(50)
WRITE(*,*)'ENTER N:'
READ(*,*)N
WRITE(*,*)'ENTER A:'
READ(*,*)(A(I),I=1,N)
! ===FILLINGOVER===
WRITE(*,*)'A=',(A(I),I=1,N)
END

```



## 参考答案

1. IF(A(K)>A(K+1)) THEN  
    D=A(K)  
    A(K)=A(K+1)  
    A(K+1)=D  
END IF
2. FUNCTION POLY(B,M,V)  
    DIMENSION B(M)  
    POLY=B(1)  
    PV=1.0  
    DO K=2,M  
        PV=PV \* V  
        POLY=POLY+B(K) \* PV  
    END DO  
END
3. DO K=3,N  
    IF(MOD(A(K),2)=-0) A(K)=0  
END DO  
ISR=SQRT(REAL(N))  
DO M=3,ISR,2  
    DO K=M+2,N  
        IF(A(K)/=0. AND. MOD(A(K),M)==0) A(K)=0  
    END DO  
END DO  
M=1
4. DO WHILE(P<=N)  
    DO WHILE(CH/=C(P). AND. P<=N)  
        P=P+1  
    END DO  
    IF(CH==C(P)) THEN  
        DO I=P,N-1  
            C(I)=C(I+1)  
        END DO  
        N=N-1

```
END IF
END DO
5. IF(X==A(MID)) THEN
    FIND=. TRUE.
    WRITE(*,100)X,'Has be found on',MID,'the position'
ELSE IF(X<A(MID)) THEN
    BOT=MID-1
ELSE
    TOP=MID+1
ENDIF
6. IF(J==0) THEN
    J=81
ELSE
    N=N+1
    J=J+5
END IF
7. IF(C(I)>10) THEN
    M=1
    C(I)=C(I)-10
ELSE
    M=0
END IF
8. DO WHILE(I<=J. AND. W==0)
    IF(MOD(A,I)==0) THEN
        W=1
    ELSE
        I=I+1
    END IF
END DO
9. DO I=2,N
    A(I-1)=A(I)
END DO
A(N)=T
10. DO WHILE(K<=N)
    DO WHILE (X/=P(K). AND. K<=N)
        K=K+1
    END DO
    IF(X==P(K)) THEN
```

```
      DO I=K,N-1
        P(I)=P(I+1)
      END DO
      N=N-1
    END IF
  ENDDO
11. FUNCTION F(N)
  F=0
  DO I=1,N
    F=F+I
  END DO
  END
12. DO WHILE(I<=65)
  K=K+I
  I=I+5
  END DO
13. DO WHILE(X<=10.0)
  S=S+X
  T=T+X*X
  X=X+0.2
  END DO
14. DO I=2,10
  IF(X(I)>XMAX) XMAX=X(I)
  IF(X(I)<XMIN) XMIN=X(I)
  END DO
15. DO WHILE(I<100)
  WRITE(*,*)I
  I=I+J
  J=I-J
  END DO
16. DO I=1,N/2
  T=A(I)
  A(I)=A(N-I+1)
  A(N-I+1)=T
  END DO
```

## 5.6 编程题

1.  $N$  个数的几何平均数定义为:  $g = \sqrt[N]{y_1 \cdot y_2 \cdot y_3 \cdot \cdots \cdot y_n}$ , 求  $[40, 46]$  之间所有偶数的几何平均值。 (答案: 42.941770)

程序如下:

```
K=0
F=1
DO I=40,46,2
    K=K+1
    F=F*I
END DO
G=F**(1.0/K)
WRITE(*,*)G
END
```

2. 有四个整数, 其和为 100, 且第一个数加 4, 第二个数减 4, 第三个数乘以 4, 第四个数除以 4 后, 它们的和、差、积、商都相等。求这四个整数。 (答案: 12, 20, 4, 64)

程序如下:

```
IMPLICIT INTEGER (X)
DO X=4,100,4
    X1=X-4
    X2=X+4
    X3=X/4
    X4=X*4
    IF(X1+X2+X3+X4, EQ, 100) THEN
        WRITE(*,*)X,X1,X2,X3,X4
    ENDIF
END DO
END
```

3. 求  $\frac{1}{1 \times 2 \times 3} + \frac{1}{2 \times 3 \times 4} + \cdots + \frac{1}{100 \times 101 \times 102}$ 。 (答案: 0.2499515)

程序如下:

```
S=0
DO N=1,100
    S=S+1.0/N/(N+1)/(N+2)
END DO
```

```
PRINT *,S
```

```
END
```

4. 求  $\sqrt{1} + \sqrt{2} + \dots + \sqrt{100}$  的值。

(答案: 671.463000)

程序如下:

```
S=0
```

```
DO N=1,100
```

```
    S=S+SQRT(N*1.0)
```

```
END DO
```

```
PRINT *,S
```

```
END
```

5. 求 100~999 之间满足下列条件的数:

(1) 该数个位数字与百位数字之和等于十位数字

(2) 该数可以被 5 整除

求所有满足条件的数的和与平均值。

(答案: 6270.482.307700)

程序如下:

```
SUM=0
```

```
K=0
```

```
DO I=100,999,5
```

```
    I1=I/100
```

```
    I2=MOD(I/10,10)
```

```
    I3=MOD(I,10)
```

```
    IF(I2.EQ.I1+I3) THEN
```

```
        K=K+1
```

```
        SUM=SUM+I
```

```
    ENDIF
```

```
END DO
```

```
WRITE(*,*)SUM, SUM/K
```

```
END
```

6. 设  $f(n) = 1 \cdot 2 \cdot 3 + 4 \cdot 5 \cdot 6 + \dots + (3n-2)(3n-1)(3n)$

(1)  $f(20) = ?$

(答案: 1114470)

(2)  $f(100) = ?$

(答案: 6.794624E+08)

程序如下:

```
F=0
```

```
READ *,N
```

```
DO I=1,N
```

```
    F=F+(3*I-2.0)*(3*I-1.0)*(3*I)
```

```
END DO
```

```
PRINT *,F
```

END

7. 求  $x=-2, x=2, y=0$  及  $y=5-x^2$  所包围的面积要求:用与  $y$  轴平行的线段将区间  $(-2, 2)$  平分为 200 等份,对每一小块面积使用梯形公式求积,并将 200 个小梯形面积之和作为所求面积的近似值。 (答案:14.666410)

程序如下:

```

F(X)=5-X * X
READ *,A,B
N=200
H=(B-A)/N
X=A
S=0
DO I=1,N
    S=S+(F(X)+F(X+H))*H/2
    X=X+H
END DO
PRINT *,S
END

```

8. 设  $f(n)$  是自然数  $n$  的各位数的平方和,如  $f(121)=1^2+2^2+1^2=6$ ,求:

(1)  $\sum_{n=100}^{200} f(n)$ ; (答案:5804)

(2) 求 100 以内满足  $f(n)=25$  的所有的  $n$ 。 (答案:5,34,43,50)

程序如下:

```

(1) INTEGER F,S
    S=0
    DO I=100,200
        F=(I/100)* * 2+MOD(I/10,10)* * 2+MOD(I,10)* * 2
        S=S+F
    END DO
    PRINT *,S
END
(2) INTEGER F
    DO I=1,100
        F=(I/100)* * 2+MOD(I/10,10)* * 2+MOD(I,10)* * 2
        IF(F.EQ.25)PRINT *,I
    END DO
END

```

9. 用欧拉筛法求 1~100 之间的素数:将 1~100 存储在一个数组中,先将 1 替换成 0,然后将除 2 以外的凡是 2 的倍数的那些数(如 4,6,8)变成 0。接下去,再将除 3 以外的

凡是 3 的倍数的那些数变成 0。如此继续下去,一直到 49,留下的没有置换成 0 的那些数便是素数。

程序如下:

```

INTEGER A(100)
DO I=1,100
  A(I)=I
END DO
A(1)=0
DO I=2,49
  DO J=I+1,100
    IF(MOD(A(J),I).EQ.0) A(J)=0
  END DO
END DO
DO I=1,100
  IF(A(I).NE.0) PRINT *,A(I)
END DO
END

```

10. 求  $\sum_{i=1}^N \frac{(i-1)}{i+1} \leq E$  的最大的 N 是 \_\_\_\_\_。

(1) 当  $E=28.8$ ;

(答案:35)

(2) 当  $E=168.8$ 。

(答案:178)

程序如下:

```

READ *,E
SUM=0
I=0
DO WHILE(SUM.LE.E)
  I=I+1
  SUM=SUM+(I-1.0)/(I+1.0)
END DO
PRINT *,I-1
END

```

11. 一个数列第 1,2,3 项分别是 1,2,3,以后每项都是它前 3 项之和。求  $\leq M$  的最大项。

(1)  $M=1280$ ;

(答案:778)

(2)  $M=20000$ 。

(答案:16377)

程序如下:

```

INTEGER A(100)
READ *,M

```

```

A(1)=1
A(2)=2
A(3)=3
N=3
DO WHILE(A(N).LE. M)
    N=N+1
    A(N)=A(N-1)+A(N-2)+A(N-3)
END DO
PRINT *,A(N-1)
END

```

12. 已知  $F_1=0, F_2=1, F_n=F_{n-1}+2F_{n-2}, (n \geq 3)$ , 求:

(1)  $F_n < 10000$  时的最大的  $n$ ;

(答案:5461)

(2) 输出  $F_{16} \sim F_{12}$ 。

(答案:171,341,683)

程序如下:

```

INTEGER F(100)
READ *,M
F(1)=0
F(2)=1
N=2
DO WHILE(F(N).LE. M)
    N=N+1
    F(N)=F(N-1)+2 * F(N-2)
END DO
PRINT *,F(N-1)
WRITE(*,*) (F(I),I=10,12)
END

```

13. 用牛顿迭代法求方程  $x^3 + \ln x - 25 = 0$  的根及迭代次数  $n$  (设迭代初值  $x_0 = 3.0$ ), 当  $|x_{n+1} - x_n| < 10^{-6}$  时停止迭代, 此时的  $x_{n+1}$  作为近似根 (保留 5 位小数)。

(答案:  $x=2.882152, n=4$ )

程序如下:

```

F(X)=X * * 3+LOG(X)-25
F1(X)=3 * X * * 2+1/X
X=3.0
N=0
DO WHILE(ABS(X-X0).GE. 1E-6)
    X0=X
    X=X0-F(X0)/F1(X0)
    N=N+1

```



```

END DO
PRINT *,X,N
END

```

14. 求  $1 + \frac{1}{2!} + \frac{1}{3!} + \cdots + \frac{1}{n!} > 1.718280$  的最小的  $n$ 。 (答案:8)

程序如下:

```

S=0
N=0
F=1.0
DO WHILE(S. LE. 1.718280)
    N=N+1
    F=F/N
    S=S+F
END DO
PRINT *,N-1
END

```

15. 计算  $s = x + \frac{x^5}{5!} + \frac{x^9}{9!} + \cdots + \frac{x^{4n+1}}{(4n+1)!}$ 。

当  $x=1.1$  时,  $s=$  \_\_\_\_\_ (精确到小数点后第 6 位)。 (答案:1.113428)

程序如下:

```

READ *,X
S=X
F=X
N=0
DO WHILE(F. GE. 1E-6)
    N=N+1
    F=F * X * * 4 / (4 * N + 1.0) / (4.0 * N) / (4 * N - 1.0) / (4 * N - 2.0)
    S=S+F
END DO
PRINT *,S
END

```

16. 设  $s = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}$ , 求与 8 最接近的  $s$  的值及与之对应的  $n$  值。

(答案:  $s=7.999890, n=1673$ )

程序如下:

```

S=0
N=0
DO WHILE((8.0-S). GE. 1E-6)
    N=N+1

```

```

    S=S+1.0/N
  END DO
  PRINT *,S-1.0/N,N-1
END

```

17. 求不定方程  $\begin{cases} x^2 - y^2 = 10000 \\ x \leq y \end{cases}$  的全部自然数解。

(答案:  $\begin{cases} x=28 \\ y=96 \end{cases} \quad \begin{cases} x=60 \\ y=80 \end{cases}$ )

程序如下:

```

  INTEGER X,Y
  DO Y=2,100
    DO X=1,Y
      IF(X*X+Y*Y.EQ.10000) PRINT *,X,Y
    END DO
  END DO
END

```

18. 若自然数  $n$  各位数之和为 12, 则称数  $n$  符合 R 条件(如 75、1650 均符合 R 条件), 求 10000 以内所有符合 R 条件的数之和 SUM 及最大、最小两数的差 SUB。

(答案: 1383195.9261)

程序如下:

```

  INTEGER RMIN,RMAX
  SUM=0
  K=0
  DO N=1,10000
    N1=N/1000
    N2=MOD(N/100,10)
    N3=MOD(N/10,10)
    N4=MOD(N,10)
    IF(N1+N2+N3+N4.EQ.12) THEN
      SUM=SUM+N
      K=K+1
      IF(K.EQ.1) RMIN=N
      RMAX=N
    END IF
  END DO
  PRINT *,SUM
  PRINT *,RMAX-RMIN
END

```

19. 求 1000~2000 之间 5 个最大的素数之和及第 5 大的素数。(答案:9955,1979)

程序如下:

```

INTEGER S(100)
K=0
DO N=1000,2000
    J=2
    M=SQRT(N*1.0)
    DO WHILE(J.LE.M.AND.MOD(N,J).NE.0)
        J=J+1
    END DO
    IF(J.GT.M) THEN
        K=K+1
        S(K)=N
    END IF
END DO
SUM=0
DO N=K,K-4,-1
    SUM=SUM+S(K)
END DO
PRINT *,SUM,S(K-4)
END

```

20. 编写程序,将输入的任意一英文单词反序输出。

程序如下:

```

CHARACTER * 20 STR,STR1
READ *,STR
N=LEN(STR)
DO I=1,N
    M=N-I+1
    STR1(I:1)=STR(M:M)
END DO
PRINT *,STR1
END

```

21. 有这样一个数列,该数列的前两项为输入的自然数,以后各项有其前面的两项之和按下列规则生成:若和小于 100,则它是下一项;若和大于 100,且最后两位数字不全为 0,则和除以 100 的商和余数依次作为下两项;否则以和除以 100 的商作为下一项。例如,若前两项为 15,17,则后面各项依次为 32,49,81,1,30,...,前 6 项之和为 195。

(1) 该数列的第 10 项;

(答案:92)

(2) 前 15 项之和。

(答案:557)

程序如下:

```

INTEGER A(100),C,SUM
A(1)=15
A(2)=17
N=2
SUM=A(1)+A(2)
DO WHILE(N, LE, 20)
    N=N+1
    C=A(N-1)+A(N-2)
    IF(C, LT, 100) THEN
        A(N)=C
    ELSE
        IF(MOD(C,10), NE, 0. OR, MOD(C/10,10), NE, 0) THEN
            A(N)=C/100
            N=N+1
            A(N)=MOD(C,100)
        ELSE
            A(N)=C/100
        END IF
    END IF
END DO
DO J=1,15
    IF(J, EQ, 10) WRITE(*,*) 'A(' ,J, ')=' ,A(J)
    SUM=SUM+A(J)
END DO
WRITE(*,*) SUM
END

```

22. 计算被 2,3,5 除,其余数均为 1 的前 10 个自然数之和,并统计其中素数的个数。

(答案:1360,7)

程序如下:

```

FUNCTION SUSHU(N)
SUSHU=0
J=SQRT(N*1.0)
I=2
DO WHILE(I, LE, J, AND, MOD(N,I), NE, 0)
    I=I+1
END DO
IF(I, GT, J) SUSHU=1

```

```

END
PROGRAM MAIN
K=0
S=0
I=1
N=0
DO WHILE(N. LT. 10)
  IF (MOD(I,2). EQ. 1. AND. MOD(I,3). EQ. 1. AND. MOD(I,5). EQ. 1) &
    THEN
    IF (I. GT. 1. AND. SUSHU(I). EQ. 1) K=K+1
    S=S+1
    N=N+1
  ENDIF
  I=I+1
END DO
WRITE(*,*) 'S=',S
WRITE(*,*) 'K=',K
END

```

23. 求 100! 的末尾有多少个 0? 1000! 的末尾有多少个 0? (答案:22,221)

程序如下:

```

L1=100
L2=1000
CALL SUB(L1,N)
WRITE(*,*) '100! HAVE 0 =',N
CALL SUB(L2,N)
WRITE(*,*) '1000! HAVE 0 =',N+1
END
SUBROUTINE SUB(L,N)
N=0
DO I=5,L,5
  N=N+1
  IF(MOD(I/10,10). EQ. 0) N=N+1
END DO
END

```

24. 定义  $f(x) = x^2 + \sqrt{1+2x+3x^2}$ , 求:

$$\alpha = (6.9+y)/(y^2 + \sqrt{1+2y+3y^2})$$

$$\beta = (2.1z+z^4)/(z^4 + \sqrt{1+2z^2+3z^4})$$

$$\Delta = 1/(\sin^2 y + \sqrt{1+2\sin y+3\sin^3 y})$$

程序如下:

```
INTRINSIC SQRT
FUN(X)=X**X+SQRT(1+2*X+3*X*X)
READ(*,*)Y,Z
ALPHA=(6.9+Y)/FUN(Y)
BETA=(2.1*Z+Z**4)/FUN(Z*Z)
DELTA=1/FUN(SIN(Y))
WRITE(*,*)ALPHA,BETA,DELTA
END
```

25. 若两个连续自然数乘积减 1 后是素数,则称此两个连续自然数为友数对,该素数称为友素数。例如: $2 \times 3 - 1 = 5$ , 2 与 3 是友数对, 5 是友素数。求 [2, 99] 之间:

(1) 友数对的数目; (答案: 48)

(2) 所有友素数的和。 (答案: 128044)

程序如下:

```
FUNCTION SUSHU(N)
SUSHU=0
J=SQRT(N*1.0)
I=2
DO WHILE(I.LE.J.AND.MOD(N,I).NE.0)
    I=I+1
END DO
IF (I.GT.J)SUSHU=1
END
SUM=0
DO I=2,98
    J=I*(I+1)-1
    IF(SUSHU(J).EQ.1)THEN
        K=K+1
        SUM=SUM+J
    ENDIF
END DO
WRITE(*,*)K,SUM
END
```

26. 设:  $S(m, n, k) = \sum_{i=-m}^n (i-k)^2$

设计一个计算 S 的函数子程序,并调用该函数子程序计算:

(1)  $S1 = \sum_{i=1}^{100} i^2$ ;

$$(2) S2 = \sum_{i=10}^{100} (i-5)^2.$$

程序如下:

```

FUNCTION MS(M,N,K)
  MS=0
  DO I=M,N
    MS=MS+(I-K)**2
  ENDDO
END
PROGRAM EX3
MS1=MS(1,100,0)
MS2=MS(10,100,5)
WRITE(*,100)MS1,MS2
100 FORMAT(5X,'S1=',I8,' S2=',I8)
END

```

27. 设计一个计算  $n!$  的函数子程序,并调用该子程序计算数  $e$  的近似值。计算公式是:

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \cdots + \frac{1}{n!} + \cdots$$

当  $n! > 1E8$  时停止计算。

程序如下:

```

RECURSIVE FUNCTION FAC(N) RESULT(FAC1)
  IF (N==1) THEN
    FAC1=1
  ELSE
    FAC1=N * FAC(N-1)
  ENDIF
END
PROGRAM EX4
S=1
I=1
FA=FAC(I)
DO WHILE (FA<1E8)
  S=S+1/FA
  I=I+1
  FA=FAC(I)
ENDDO
WRITE(*,100)S,I-1
100 FORMAT(5X,'S=',F11.8,' N=',I3)

```

```

ENDWRITE(*,100)S,1
100 FORMAT(5X,'S=',F11.8,' N=',I3)
END

```

$$28. \text{ 已知 } F(x) = \frac{e^x - |x-12|}{x+10\cos(x-1)}, G(x) = \begin{cases} \frac{1}{x^2} & x < 9 \\ 0 & x = 9 \\ \ln(x^2+5) & x > 9 \end{cases}$$

$$\text{求 } s = \sum_{i=1}^4 \frac{4F(t^i)}{F(\frac{t}{2}-1)-3G(t^2+2)} \quad (\text{答案: } -233428.500000)$$

程序如下:

```

F(X)=(EXP(X)-ABS(X-12))/(X+10*COS(X-1))
S=0
DO T=1,0,4,0,1,0
    S=S+4*F(T*T)/(F(T/2-1)-3*G(T*T+2))
END DO
PRINT*,S
END
FUNCTION G(X)
IF(X.LT.9,0) THEN
    G=1/(X*X)
ELSEIF(X.EQ.9) THEN
    G=0
ELSE
    G=LOG(X*X+5)
END IF
END

```

$$29. \text{ 设函数 } m(x) \text{ 定义为 } m(x) = \begin{cases} x-10 & x > 100 \\ e^{x/10} & x \leq 100 \end{cases}, \text{ 求:}$$

$$(1) m(m(102));$$

(答案: 9897.129000)

$$(2) \sum_{n=21}^{30} m(n);$$

(答案: 133.418800)

程序如下:

```

REAL FUNCTION M(X)
IF(X.GT.100) THEN
    M=X-10.0
ELSE
    M=EXP(X/10.0)
END IF

```



```

END
REAL M
PRINT *, M(M(102, 0))
S=0
DO N=21,30
    S=S+ M(N * 1, 0)
END DO
PRINT *, S
END

```

30. 编写函数计算并返回  $m$  行 6 列整型数组  $A$  周边元素之和  $n$  (即第一行、第  $m$  行、第 1 列、第 6 列上元素之和, 但是重复元素只参加一次求和)。数组名  $A$  和  $m, n$  作为函数参数。

程序如下:

```

SUBROUTINE SUM(A, M, N)
INTEGER A(M, 6)
N=0
DO I=2, 5
    N=N+A(1, I)+A(M, I)
END DO
DO I=1, M
    N=N+A(I, 1)+A(I, 6)
END DO
END

```

31. 设函数  $m(x)$  定义为  $m(x) = \begin{cases} x-10 & x > 1100 \\ m(m(x+11)) & x \leq 100 \end{cases}$ , 求:

(1)  $m(100)$ ;

(答案: 91)

(2)  $\sum_{n=11}^{20} m(n)$ 。

(答案: 910)

程序如下:

```

RECURSIVE FUNCTION M(X) RESULT(K)
INTEGER X
IF(X.GT. 100) THEN
    K=X-10
ELSE
    K=M(M(X+11))
END IF
END
INTEGER X, S

```

```

X=100
PRINT *,M(X)
S=0
DO N=21,30
    S=S+M(N)
END DO
PRINT *,S
END

```

32. 对已知自然数  $n$ , 设计一个判断  $n$  是否为素数的子程序, 并调用该子程序:

(1) 将 100~120 共 11 个偶数分解为两个素数和的形式

(2) 用 READ 语句读入一个自然数  $n$ , 对  $n$  进行素因子分解。

要求使用函数子程序和子例行程序两种方法。

使用函数子程序的参考程序与运行结果如下:

```

FUNCTION PRIME(N)
INTEGER PRIME
PRIME=0
DO I=2,N-1
    IF (MOD(N,I) == 0) GOTO 10
ENDDO
PRIME=1
10 END
(1)
PROGRAM EX5
INTEGER PRIME
DO I=100,120,2
    DO J=2,I-2
        IF (PRIME(J) == 1 .AND. PRIME(I-J) == 1) THEN
            WRITE(*,100)J,I-J,I
            EXIT
        END IF
    ENDDO
END DO
100 FORMAT(5X,I3,'+',I3,'=',I3)
END

```

运行结果如下:

3+ 97=100

5+ 97=102

3+101=104

$$3+103=106$$

$$5+103=108$$

$$3+107=110$$

$$3+109=112$$

$$5+109=114$$

$$3+113=116$$

$$5+113=118$$

$$7+113=120$$

(2) 参考程序及运行结果如下:

```
PROGRAM EX5
```

```
INTEGER PRIME,A(10) ! 设一般整数的素数因子数不超过 10, 不尽合理。
```

```
WRITE(*,*)'PLEASE INPUT A INTEGER NUMBER'
```

```
READ(*,*)N
```

```
M=N
```

```
J=1
```

```
DO WHILE (PRIME(M) == 0)
```

```
DO I=2,M-1 ! 寻找 M 的一个素因子
```

```
IF (MOD(M,I) == 0 . AND. PRIME(I) == 1) THEN
```

```
    A(J)=I
```

```
    M=M/I
```

```
    J=J+1
```

```
    EXIT
```

```
ENDIF
```

```
ENDDO
```

```
ENDDO
```

```
A(J)=M
```

```
WRITE(*,100)N,(A(I),I=1,J)
```

```
100 FORMAT(4X,I6,'=1',10(' * ',I2))
```

```
END
```

第一次运行结果如下:

```
PLEASE INPUT A INTEGER NUMBER
```

```
100✓
```

$$100=1 * 2 * 2 * 5 * 5$$

第一次运行结果如下:

```
PLEASE INPUT A INTEGER NUMBER
```

```
30000✓
```

$$30000=1 * 2 * 2 * 2 * 2 * 3 * 5 * 5 * 5 * 5$$

使用子例行程序的参考答案略。

33. 对任意自然数  $n$ , 设计一个求  $n$  的各数位立方和的函数子程序  $F(n)$  (如  $F(121) = 1^3 + 2^3 + 1^3 = 1 + 8 + 1 = 10$ ), 并调用  $F(n)$  求 1000 以内的所有水仙花数。

程序如下:

```

    RECURSIVE FUNCTION NF(N) RESULT(F)
    INTEGER F
    M=N
    IF (M>9) THEN
        F=MOD(M,10)**3+NF(M/10)
    ELSE
        F=M**3
    ENDIF
    END
    PROGRAM EX6
    DO I=1,1000
        IF (NF(I)==I) THEN
            WRITE(*,*)I
        ENDIF
    ENDDO
    END

```

34. 自己设计一个求余函数  $\text{MOD\_SELF}(M,N)$ , 功能与内在函数  $\text{MOD}$  相同(不能调用  $\text{MOD}$  函数)。

函数子程序如下:

```

    FUNCTION MOD_SELF(M,N)
    M1=M
    DO WHILE (M1>N)
        M1=M1-N
    ENDDO
    IF (M1<N) THEN
        MOD_SELF=M1
    ELSE
        MOD_SELF=0
    ENDIF
    END

```

35. 设计一个函数子程序, 对于自然数  $m, n$ , 该函数求  $m, n$  的最小公倍数。

程序如下:

```

    INTEGER FUNCTION GCD(M,N)
    INTEGER R
    M1=M

```

```

N1=N
IF (M1<N1) THEN
    R=M1
    M1=N1
    N1=R
ENDIF
R=MOD(M1,N1)
DO WHILE (R>0)      ! 辗转相除
    M1=N1
    N1=R
    R=MOD(M1,N1)
ENDDO
GCD=M * N/N1        ! M、N 的乘积除以 M、N 的最大公约数
END

```

这里按题目要求未给出对该子程序的调用程序单位的代码,读者可以自己补充一段代码并上机运行。

36. 设计一个函数子程序将一个数字型字符转化为一个与其相同的数值型数据,并调用该子程序完成:

- (1) 将字符型数据 '23456' 转化为整型数据 23456 ;
- (2) 将字符型数据 '75.8' 转化为实型数据 75.8。

程序如下:

```

FUNCTION TRAN(A)
CHARACTER * 15 A
F1=0
J=0      ! 无小数位,J=0;1 位小数,J=10;2 位小数,J=100...
DO I=1,15
    IF (ICHAR(A(I:1))=0)EXIT      ! 遇空字符跳出循环
    IF (A(I:1)/='.' ) THEN
        F1=F1 * 10+ICHAR(A(I:1))-48      ! 未考虑小数位
        J=J * 10
    ELSE
        J=1
    ENDIF
ENDDO
TRAN=F1
IF (J>0) TRAN=F1/J      ! J>0,即有小数时,转换结果除以 J
END FUNCTION TRAN
PROGRAM EX9

```

```
WRITE(*,*)TRAN('23456')
WRITE(*,*)TRAN('75.8')
END
```

37. 设计一个子例行程序,计算一个任意数组 A 中正数、负数和 0 的个数及其和。  
程序如下:

```
SUBROUTINE COUNT(A,N,PLUS,NATIVE,ZERO)
! 统计数组 A 中正元素的个数放到 PLUS,负元素的个数放到 NATIVE,
! 0 元素的个数放到 ZERO
INTEGER N,PLUS,NATIVE,ZERO
DIMENSION A(N)
PLUS=0
NATIVE=0
ZERO=0
DO I=1,N
  IF (A(I)>0) THEN
    PLUS=PLUS+1
  ELSE IF (A(I)==0) THEN
    ZERO=ZERO+1
  ELSE
    NATIVE=NATIVE+1
  ENDIF
ENDDO
END SUBROUTINE COUNT
```

38. 设计一个子例行程序 SORT(A,N,K),其中 A 是一个一维数组,N 是 A 的元素个数,SORT 的功能是:当 K=1 时,将数组 A 按升序排列;当 K=0 时,将数组 A 按降序排列;当 K 为其他数值时,数组 A 保持原序。

程序如下:

```
SUBROUTINE SORT(A,N,K)
REAL T,A(N)
SELECT CASE(K)
  CASE (1)
    K1=1      ! 升序排序
  CASE (0)
    K1=-1     ! 降序排序
  CASE DEFAULT
    K1=0      ! 保持原序
END SELECT
DO I=1,N-1
```

```

      DO J=I+1,N
        IF (A(I) * K1 > A(J) * K1) THEN
          T=A(I)
          A(I)=A(J)
          A(J)=T
        ENDIF
      ENDDO
    ENDDO
  END

```

39. Ackermann 函数的定义是:

$$A(0, n) = n + 1$$

$$A(m, 0) = A(m-1, 1)$$

$$A(m, n) = A(m-1, A(m, n-1))$$

其中  $m, n$  是自然数, 设计一个计算 Ackermann 的函数子程序, 并计算  $A(5, 8)$  的值。

程序如下:

```

      RECURSIVE FUNCTION NA(M,N) RESULT(NA1)
      IF (M==0) THEN
        NA1=N+1
      ELSE IF (N==0) THEN
        NA1=NA(M-1,1)
      ELSE
        NA1=NA(M-1,NA(M,N-1))
      END IF
    END FUNCTION
  PROGRAM EX12
    WRITE(*,*) NA(5,8)
  END

```

40. 设计一个函数子程序  $\text{line}(z1, z2, z3)$ , 参数  $z1, z2, z3$  是复数, 代表平面上三个点, 当三个点共线时, 函数值为 1, 否则为 0。

函数子程序如下:

```

      FUNCTION LINE(Z1,Z2,Z3)
      COMPLEX Z1,Z2,Z3,Z1Z2,Z1Z3
      Z1Z2=Z1-Z2      ! 由 Z1、Z2 点组成的向量
      Z1Z3=Z1-Z3      ! 由 Z1、Z3 点组成的向量
      X1=REAL(Z1Z2)    ! 取 Z1Z2 的 X 分量
      Y1=AIMAG(Z1Z2)   ! 取 Z1Z2 的 Y 分量
      X2=REAL(Z1Z3)
      Y2=AIMAG(Z1Z3)

```

```

IF (X1 * Y2 - X2 * Y1) THEN      ! 向量共线条件
    LINE=1
ELSE
    LINE=0
END IF
END

```

41. 对于任意的二维数组  $A(m,n)$ , 设计一个子例行程序  $\max(A,B,m,n,k)$ 。其中  $A$  是一个二维数组,  $m,n$  分别是  $A$  的行数和列数,  $B$  是一个一维数组。子程序的功能是: 当参数  $k=1$  时, 求  $A$  的每列上的最大元素并存放到  $B(1), B(2), \dots, B(n)$  中; 当参数  $k=2$  时, 求  $A$  每行上的最大元素并存放到  $B(1), B(2), \dots, B(m)$  中。

程序如下:

```

SUBROUTINE SMAX(A,B,M,N,K)
DIMENSION A(M,N),B(MAX(M,N))
IF (K==1) THEN
    DO J=1,N
        B(J)=A(1,J)
        DO L=2,M
            IF (A(L,J)>B(J)) B(J)=A(L,J)
        END DO
    END DO
ELSE
    DO J=1,M
        B(J)=A(J,1)
        DO L=2,N
            IF (A(J,L)>B(J)) B(J)=A(J,L)
        END DO
    END DO
END IF
END

```

42. 设计一个子例行程序, 将一个字符型数据翻译成密文, 翻译规则是:

当对应字符是一个英文字母时, 将  $A \rightarrow Z(a \rightarrow z), B \rightarrow Y, C \rightarrow X, \dots$ 。

当对应字符非英文字母时, 保留该字符原文。如将字符型数据 'word!' 应翻译 'dliw!'。

子程序如下:

```

SUBROUTINE EX16(A)
CHARACTER A * 80
INTEGER MCHA
DO I=1,80

```



```

MCHA=ICHAR(A(I:I))
IF (MCHA>=65 .AND. MCHA<=90) THEN ! A(I:I)是大写字母
  A(I:I)=CHAR(155-ICHAR(A(I:I))) ! 155=ICHAR('A')+ICHAR('Z')
ELSE IF (MCHA>=97 .AND. MCHA<=122) THEN
  A(I:I)=CHAR(219-ICHAR(A(I:I))) ! 219=ICHAR('a')+ICHAR('z')
END IF
END DO
END

```

43. 设计一个矩形方法计算函数定积分的函数子程序。矩形积分公式是：

$$\int_a^b f(x)dx = h \sum_{i=0}^{n-1} f(a+i \cdot h)$$

式中： $h=(b-a)/n$ ，即将积分区间 $[a,b]$ 划分为 $n$ 个等长的区间， $n$ 是小区间的个数，而 $h$ 是小区间的长度。

函数子程序的如下：

```

FUNCTION EX17(F,A,B,N)
! F 是积分函数,A,B 是积分上下限,N 子是区间数
EX17=0
H=(B-A)/N
DO I=0,N-1
  EX17=EX17 + F(A+I * H)
ENDDO
EX17=H * EX17
END

```

44. 设计一个生成 $(0,1)$ 中随机数的函数子程序。

程序如下：

```

FUNCTION EVEN(N)
PARAMETER(K=219)
INTEGER SEED
DATA SEED/29/
SAVE SEED
DO I=2,N
  SEED=MOD(K * SEED,1024)
ENDDO
EVEN=SEED/1024.0
END FUNCTION EVEN

```

45. 有 52 张扑克牌排列为一行，并用 1~52 给每张扑克牌编号。首先将全部扑克牌正面朝上并按下列操作翻转扑克牌（即将原来正面朝上的翻转为正面朝下，将原来正面朝下扑克牌的翻转为正面朝上）。

- (1) 第一轮将 2 的倍数位置上的扑克全部翻转；  
 (2) 第二轮将 3 的倍数位置上的扑克牌全部翻转，…；  
 (3) 重复上述过程直到最后一轮将编号为 52 的扑克牌翻转。  
 设计一个递归子例行程序模拟上述过程，并指出最后正面朝上的扑克牌位置，  
 程序如下：

```

RECURSIVE SUBROUTINE PLAY_CARD(A,M,N)
INTEGER A(M)
IF (N<=M) THEN
  DO I=N,M
    IF (MOD(I,N) == 0) A(I) = MF(A(I))
  ENDDO
  CALL PLAY_CARD(A,M,N+1)
ENDIF
END SUBROUTINE PLAY_CARD
FUNCTION MF(N)
  SELECT CASE(N)
    CASE(0)
      MF=1
    CASE(1)
      MF=0
  END SELECT
END FUNCTION MF
PROGRAM PLAY_CARD_PRO
PARAMETER(M=52)
INTEGER A(M)
DATA A/M*1/      ! 设开始扑克牌全部正面朝上
CALL PLAY_CARD(A,M,2)
WRITE(*,100)A
100 FORMAT(8I9)
END

```

46. 输出指定文件“FILE1.TXT”中首字母为 A(无论大小写)的那些词。  
 程序如下：

```

CHARACTER CH*1,WORD*20
OPEN(8,FILE="FILE1.TXT",ACCESS="SEQUENTIAL")
DO WHILE(.NOT.EOF(8))
  I=0
  READ(8,'(A)',ADVANCE="no") CH
  DO WHILE(CH.NE." ")

```

```

      I=I+1
      WORD(1:1)=CH
      READ(8,'(A)',ADVANCE="no",EOR=10) CH
    END DO
    IF(WORD(1:1).EQ."A".OR. WORD(1:1).EQ."a") PRINT *,WORD
10 WORD=""
  END DO
END

```

47. 建立一个直接文件,保存网站用户资料,包括姓名、详细地址和邮编。  
程序如下:

```

      CHARACTER NAME * 10 ,ADDRESS * 30,ZIPCODE * 6
      OPEN (8,FILE="INFO.TXT",ACCESS="DIRECT",FORM=&
           "FORMATTED",RECL=55)
      I=0
      READ *, NAME, ADDRESS, ZIPCODE
      DO WHILE(NAME.NE. "!")
        I=I+1
        WRITE(8,100,REC=I) NAME, ADDRESS, ZIPCODE
        READ *, NAME, ADDRESS, ZIPCODE
      END DO
100 FORMAT(A10,3X,A30,3X,A6)
      END

```

48. 从键盘输入一个省份,输出上题建立的文件中该省所有用户的资料。  
程序如下:

```

      CHARACTER NAME * 10 ,ADDRESS * 30,ZIPCODE * 6 ,P * 6
      OPEN (8,FILE="INFO.TXT",ACCESS="DIRECT",FORM=&
           "FORMATTED",RECL=55)
      READ *,P
      N=LEN(P)
      I=1
      READ(8,100,REC=1) NAME, ADDRESS, ZIPCODE
      DO WHILE(.NOT. EOF(8))
        IF(INDEX(ADDRESS,P(1:N))) PRINT *, NAME, ADDRESS, ZIPCODE
        NAME=""
        ADDRESS=""
        ZIPCODE=""
        I=I+1
        READ(8,100,REC=I) NAME, ADDRESS, ZIPCODE
      END DO
100 FORMAT(A10,3X,A30,3X,A6)
      END

```

```

END DO
100 FORMAT(A10,3X,A30,3X,A6)
200 FORMAT(1X,A10,3X,A30,3X,A6)
END

```

49. 绘制阿基米德螺旋线。其参数方程为：

$$\begin{cases} x=n * \sin n \\ y=n * \cos n \end{cases}$$

程序如下：

```

SUBROUTINE DRAW_FORM()
USE MSFLIB
INTEGER(4) COLOR,STATUS
TYPE (QWINFO) WINFO,QWI
TYPE (XYCOORD) ORIGIN
PARAMETER (PI=3.141593)
QWI. TYPE=QWIN$ MAX
STATUS=SETWSIZEQQ(QWIN$ FRAMEWINDOW,QWI)
OPEN(10,FILE='USER',TITLE="Draw_Form" )
WINFO. TYPE=QWIN$ MAX
STATUS=SETWSIZEQQ(10,WINFO)
CALL SETVIEWORG(320,240,ORIGIN)
COLOR= #0000FF
DO A=0,12,0.01
    X=A * SIN(A) * 10
    Y=A * COS(A) * 10
    STATUS=SETPIXELRGB(X,Y,COLOR)
END DO
END
CALL DRAW_FORM()
END

```