云南大学数学与统计学实验教学中心  
《高级语言程序设计》实验报告

|  |  |  |
| --- | --- | --- |
| 课程名称：程序设计和算法语言 | 学期：2016~2017学年上学期 | 成绩： |
| 指导教师：赵越 | 学生姓名：刘鹏 | 学生学号：20151910042 |
| 实验名称：简单程序设计调试 |  |  |
| 实验编号：No.01 | 实验日期：2017年1月16日 | 实验学时：2 |
| 学院：数学与统计学院 | 专业：信息与计算科学 | 年级：2015级 |

# 实验目的

1. 熟悉TC或Borland C++的运行环境。
2. 了解在TC或Borland C++上如何编辑、编译、连接和运行一个C程序。
3. 熟练使用各种C语言的数据类型、运算符与表达式。
4. 熟练掌握格式输入与格式输出函数的使用。
5. 通过运行简单的Ｃ程序，初步了解Ｃ源程序的特点,初步掌握查错和排错的技能.
6. 学会Turbo C 2.0的安装方法（参看本书第一部分），熟悉C语言程序的运行环境，了解所用计算机系统软、硬配置。
7. 初步了解在该集成环境下如何编辑、编译、连接和运行一个C程序，即运行一个C程序的全过程。
8. 通过运行简单的C程序，初步了解C程序的基本结构及特性。

# 实验环境

Windows10 Pro Workstation 17096；

Code::Blocks 16.01 GCC集成开发环境；

Cygwin GCC编译器。

# 实验内容

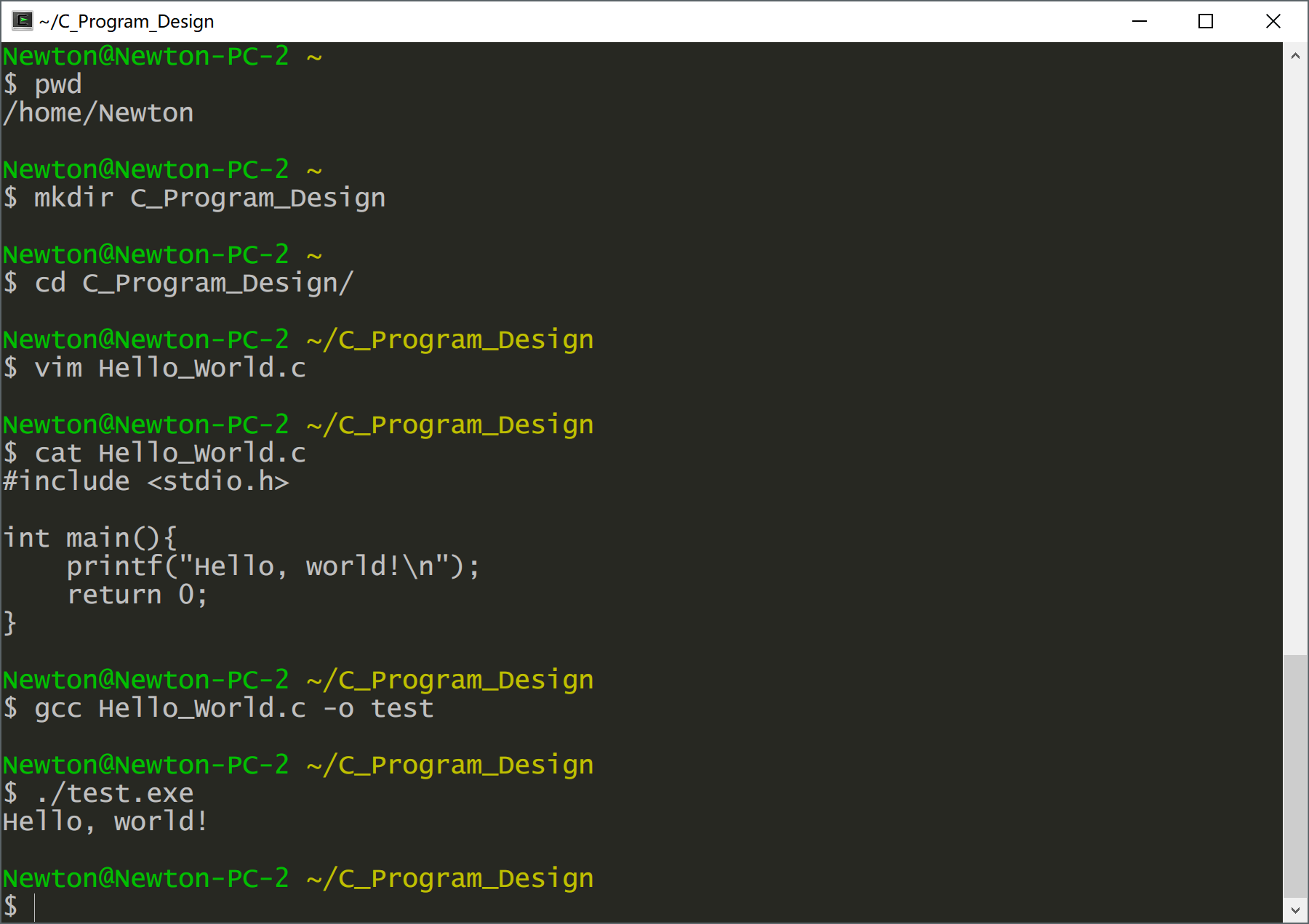
## 题1

从开机开始进行操作，熟悉一些常用的DOS命令，包括如何建立子目录，拷贝文件，删除文件等。

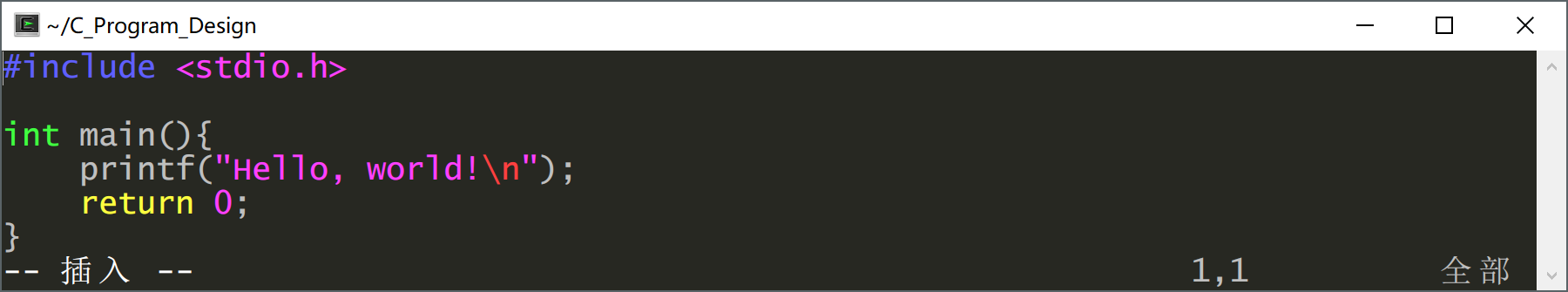
**Solution**

标准C总是与UNIX系统关联在一起，为了使用符合国际标准的C编译器，这里引入Cygwin这套软件，它可以提供一个符合POSIX标准的运行时环境，同时使用很多UNIX环境下的软件，比如这里使用的GCC编译器与Vim编辑器。

包括本篇在内所有实验报告均采用时下最新的GCC编译器进行编译，其他组织或者企业实现的编译器，如Visual C++等均不在采用之列。也可以采用MinGW，它主要提供了针对win32应用的GCC、GNU binutils等工具，以及对等于Windows SDK的头文件和用于MinGW版本linker的库文件，而不是Visual C的lib。



其中，Vim输入部分为



## 题2

建立自己的子目录，以备存放文件。

**Solution**

根据上一题，已经建立了一个文件夹用来存放代码。以后的代码文件名将与题目联系。这里用命令

mv Hello\_World.c "1.1 Hello World.c"

进行了重命名。生成的可执行文件根据需要进行及时清理。

## 题3

进入Turbo C集成环境，熟悉Turbo C主菜单下各选择项的功能及功能键的使用。

Turbo C是美国Borland公司的产品，Borland公司是一家专门从事软件开发、研制的大公司。该公司相继推出了一套 Turbo系列软件, 如Turbo BASIC, Turbo Pascal, Turbo Prolog, 这些软件很受用户欢迎。该公司在1987年首次推出Turbo C 1.0 产品，其中使用了全然一新的集成开发环境，即使用了一系列下拉式菜单，将文本编辑、程序编译、连接以及程序运行一体化，大大方便了程序的开发。1988年, Borland 公司又推出Turbo C1.5版本，增加了图形库和文本窗口函数库等，而Turbo C 2.0 则是该公司1989年出版的。Turbo C2.0在原来集成开发环境的基础上增加了查错功能，并可以在Tiny模式下直接生成.COM（数据、代码、堆栈处在同一）64K内存中文件。还可对数学协处理器(支持8087/80287/80387等)进行仿真。

Borland 公司后来又推出了面向对象的程序软件包Turbo C++，它继承发展Turbo C 2.0 的集成开发环境, 并包含了面向对象的基本思想和设计方法。1991年为了适用Microsoft 公司的Windows 3.0 版本, Borland 公司又将Turbo C++ 作了更新, 即Turbo C的新一代产品Borland C++也已经问世了。很多国家也在使用和接受中。2006年，Embarcadero Technologies将Turbo C与MS-DOS版本的Turbo C++释出成为自由软件。

|  |  |
| --- | --- |
| （1）启动TC集成环境 | （2）编辑一个新文件    按F10将亮条移到File处按回车键，再选New后按回车键，为简化，以后用File→New的形式表示上述操作。 |
| （3）文件命名    出现子窗口后，在其中输入要创建的文件名，输入文件名后按回车键，出现编辑窗口，即可在此窗口中输入和修改源程序。 | （3）保存文件 |
| （4）编译文件    按F10，选择菜单Compile→Compile to obj，就可以对程序进行编译。在此过程中可能会出现错误信息，并进行修改，修改完后存盘再进行编译，直至成功。 | （5）运行文件    按F10，选择Run→Run命令运行文件。 |
| （6）查看运行结果    按F10，选择菜单Run→User screen即可以查看程序运行结果。注意：若编译没有成功，则程序不能运行；若编译成功，则可以运行程序，但若运行结果不正确，则表明程序有逻辑错误，需要对程序进行调试和修改，直至程序运行结果正确。 | （7）学会调入一个已存在的C源文件    按F10，选择菜单File→Load，出现子窗口，在其中输入已存在的文件名，按回车键后即可以调入一个已存在的文件。 |
| （8）学会退出Turbo C    暂时退出：按F10，选择菜单File→OS shell。此时按exit即可以返回主菜单。 | （9）永久退出：按F10，选择菜单File→Quit。 |

## 题4

输入一个简单C程序（可用教科书上的例题），了解C程序运行的全过程。

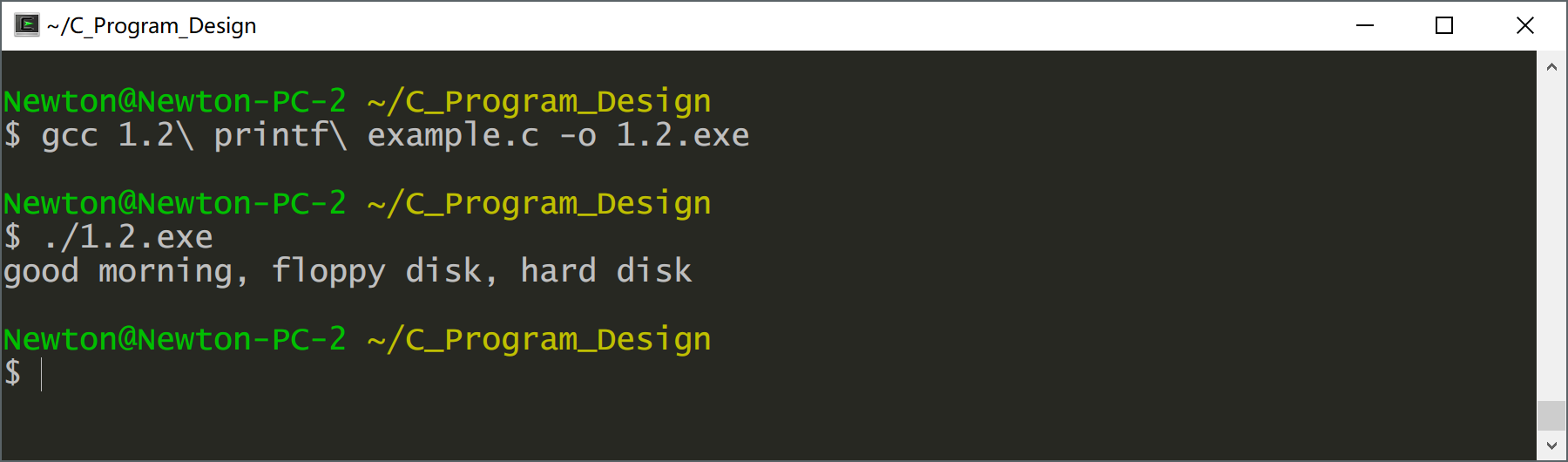
## 题5

编写用printf语句将3个字符串：good morning，floppy disk，hard disk在同一行显示的程序。

### 程序示例

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | /\*  \* filename: 1.2 example print.c  \* property: example  \*/  #include<stdio.h>  int main**()** **{**  printf **(**"good morning, "**);**  printf **(**"floppy disk, "**);**  printf **(**"hard disk\n"**);**  **return** 0**;**  **}** |

### 运行结果



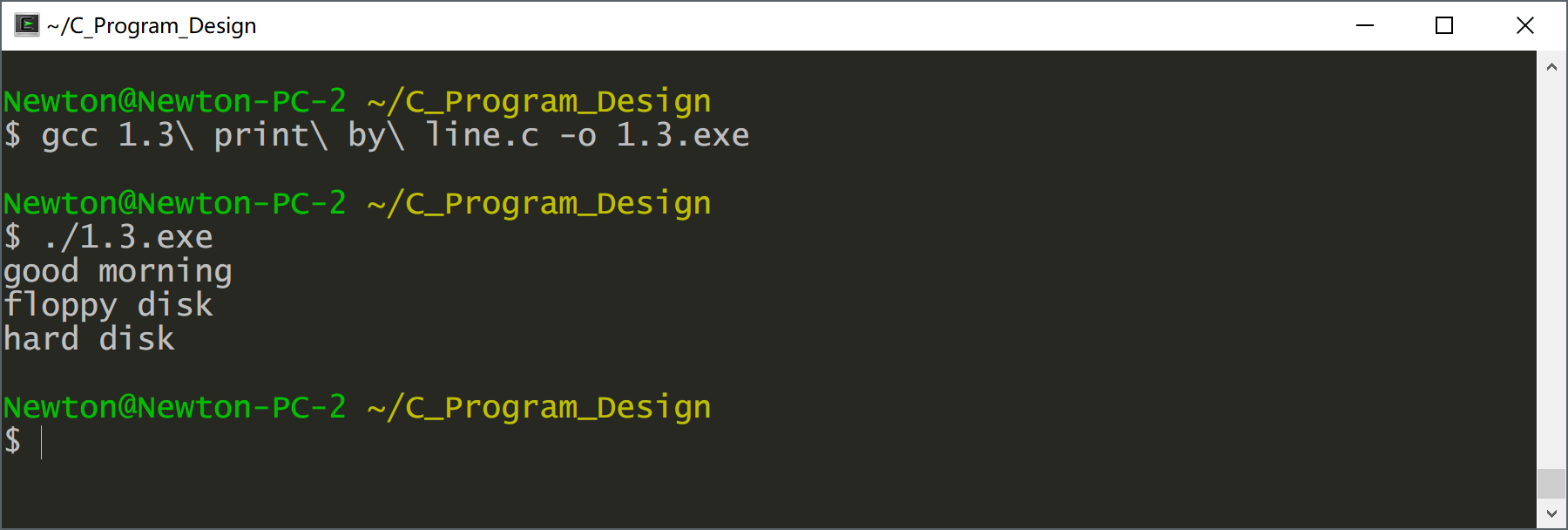
## 题6

把上面的程序改为每行显示一个字串，应如何修改程序，并运行之。

### 程序代码

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | /\*  \* filename: 1.3 print by line.c  \* property: example  \*/  #include<stdio.h>  int main**()** **{**  printf **(**"good morning\n"**);**  printf **(**"floppy disk\n"**);**  printf **(**"hard disk\n"**);**  **return** 0**;**  **}** |

### 运行结果



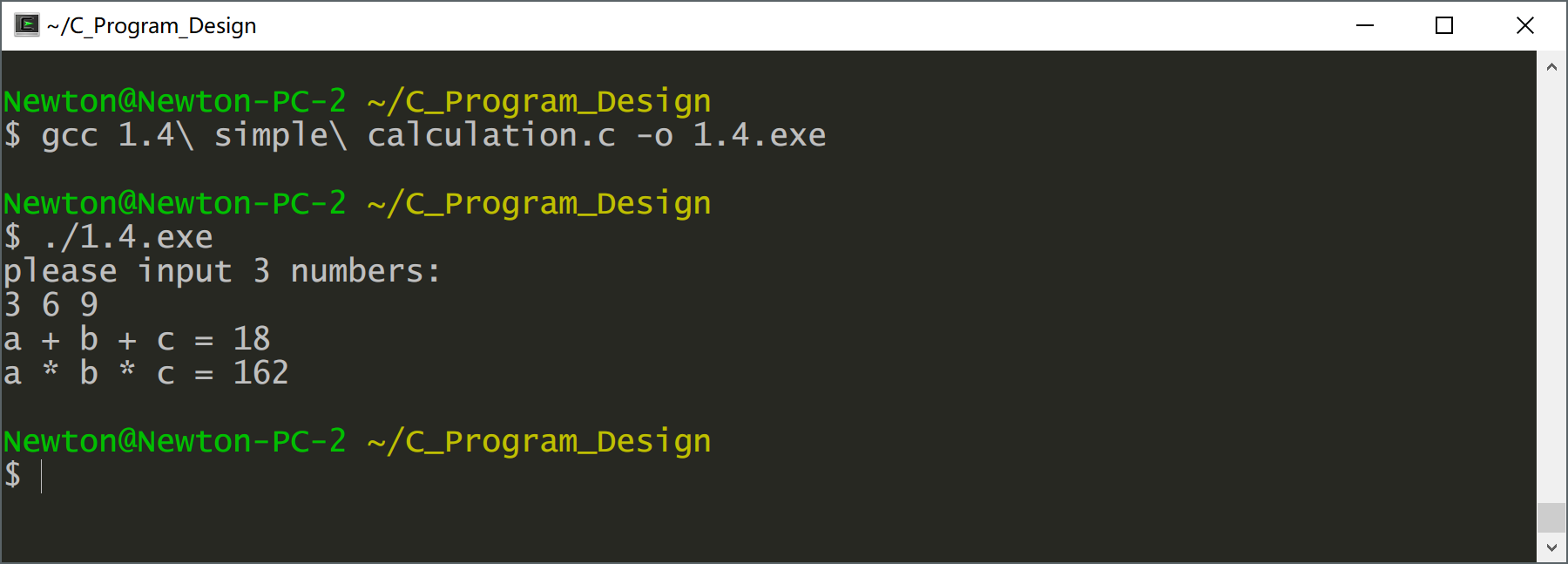
## 题7

编写一程序，用键盘输入语句输入三个数，然后分别求它们的和与积。

### 程序代码

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | /\*  \* filename: 1.4 simple calculation.c  \* property: homework  \*/  #include<stdio.h>  int main**()** **{**  int x**,** y**,** z**;**  printf**(**"please input 3 numbers:\n"**);**  scanf**(**"%d %d %d"**,** **&**x**,** **&**y**,** **&**z**);**  printf**(**"a + b + c = %d\n"**,**x**+**y**+**z**);**  printf**(**"a \* b \* c = %d\n"**,**x**\***y**\***z**);**  **return** 0**;**  **}** |

### 运行结果



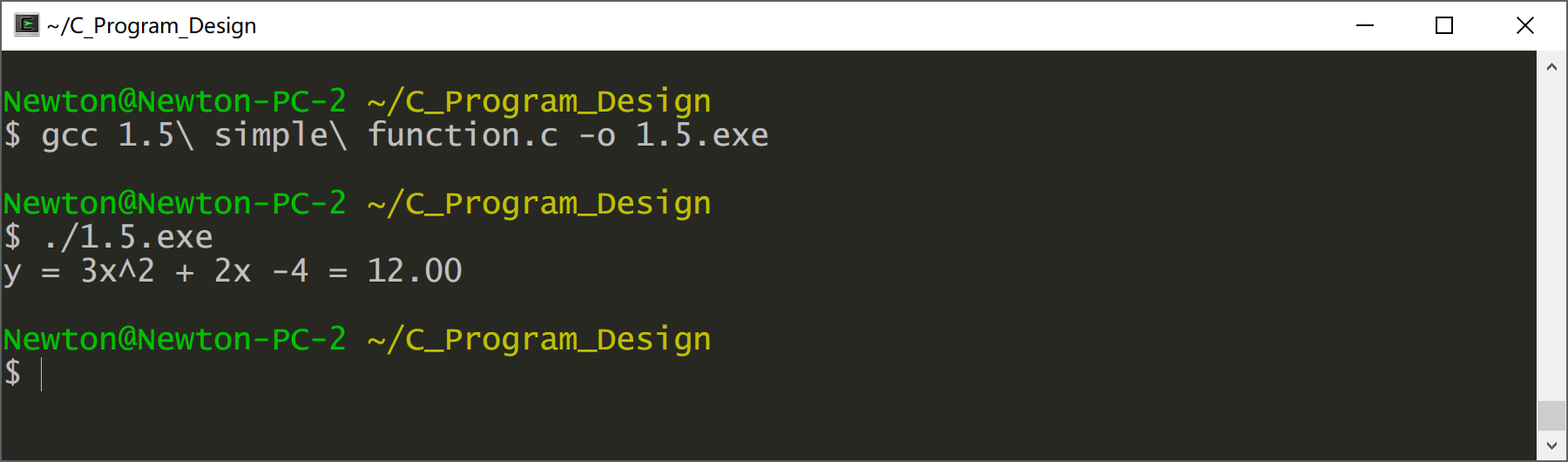
## 题8

计算函数（假设）。

### 程序代码

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | /\*  \* filename: 1.5 simple function.c  \* property: example  \*/  #include <stdio.h>  #include <math.h>  int main**()** **{**  double x **=** 2**;**  double y**;**  y **=** 3 **\*** pow**(**x**,** 2**)** **+** 2 **\*** x **-** 4**;**  printf**(**"y = 3x^2 + 2x -4 = %1.2f\n"**,** y**);**  **return** 0**;**  **}** |

### 运行结果



## 题9

设一个正圆台的上底半径，下底半径，高。分别计算并输出该圆台的上底面积下底面积，侧面积，圆台的体积。

具体要求如下:

（1），，用scanf函数输入，且在输入前要有提示；

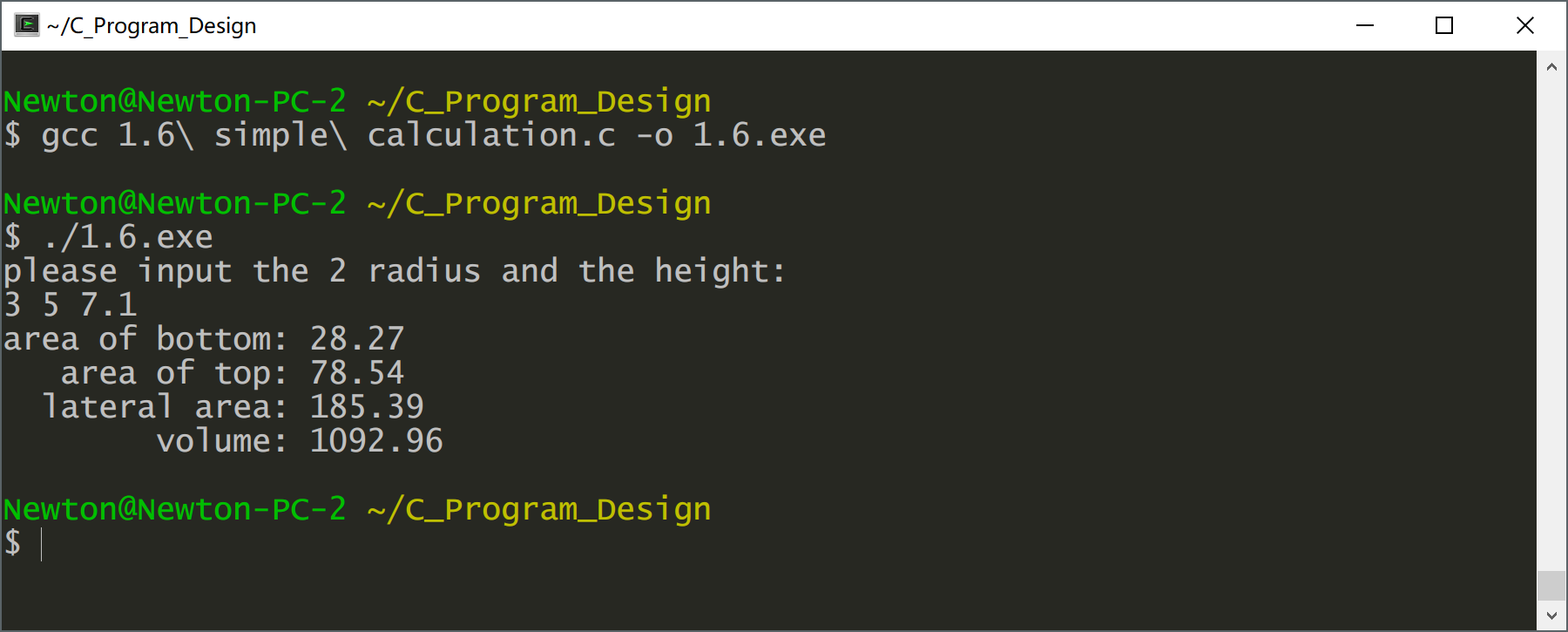
（2）在输出结果时要有文字说明，每个输出值占一行，且小数点后取4位数字。

（3）所有变量均定义为单精度类型。

### 程序代码

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | /\*  \* filename: 1.6 simple calculation.c  \* property: homework  \*/  #include <stdio.h>  #include <math.h>  #define Pi 3.14159265  int main**()** **{**  float r1**,** r2**,** h**,** S1**,** S2**,** S3**,** V**;**  printf**(**"please input the 2 radius and the height:\n"**);**  scanf**(**"%f %f %f"**,&**r1**,** **&**r2**,** **&**h**);**  S1**=**Pi **\*** pow**(**r1**,** 2**);**  S2**=**Pi **\*** pow**(**r2**,** 2**);**  S3**=**Pi **\*** **(**r1 **+** r2**)** **\*** sqrt**(**pow**(**r1 **-** r2**,** 2**)** **+** pow**(**h**,** 2**));**  V**=**Pi **\*** h **\*** **(**pow**(**r1**,** 2**)** **+** pow**(**r2**,** 2**)** **+** r1 **\*** r2**);**  printf**(**"area of bottom: %-3.2f\n"**,** S1**);**  printf**(**" area of top: %-3.2f\n"**,** S2**);**  printf**(**" lateral area: %-3.2f\n"**,** S3**);**  printf**(**" volume: %-3.2f\n"**,** V**);**  **return** 0**;**  **}** |

### 运行结果



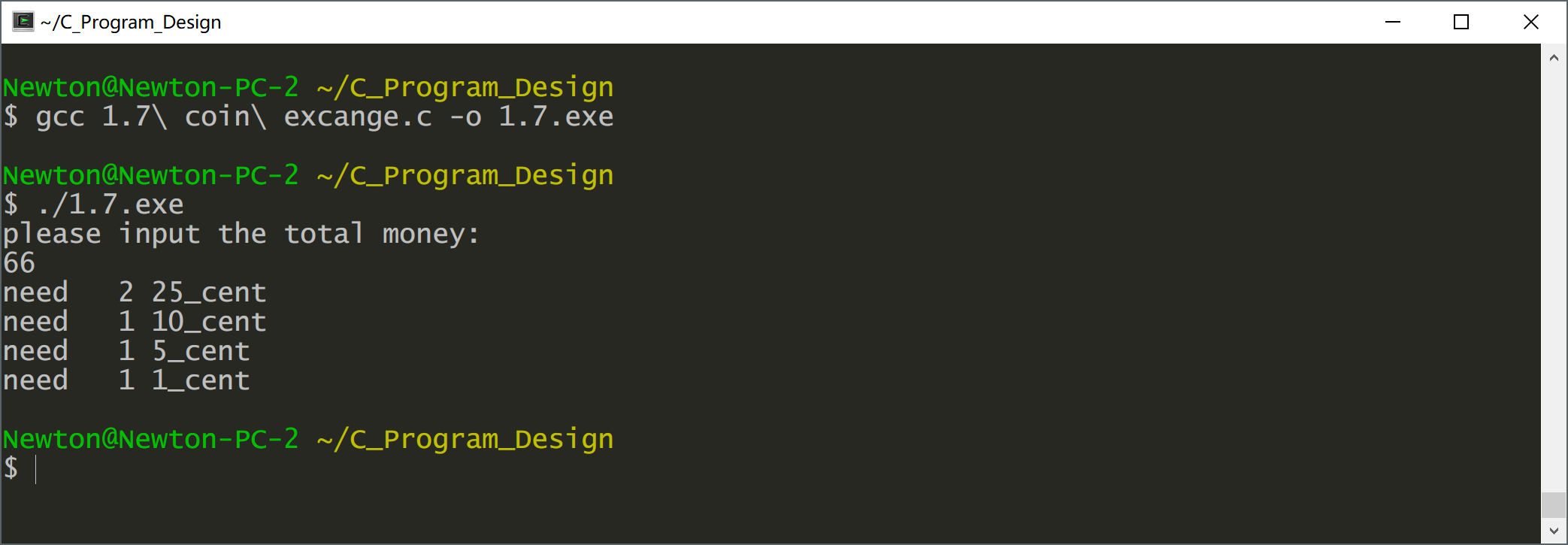
## 题10

编程实现：用1分、5分、10分、25分的硬币凑成任何给定的钱数，要求硬币的数目最少。

### 程序代码

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | /\*  \* filename: 1.7 coin exchange.c  \* property: homework  \*/  #include<stdio.h>  int main**()** **{**  int total**;**  int k\_1**,**k\_2**,**k\_3**,**k\_4**;**  printf**(**"please input the total money:\n"**);**  scanf**(**"%d"**,** **&**total**);**  k\_1 **=** total **/** 25**;**  total **=** total **-** k\_1 **\*** 25**;**  k\_2 **=** total **/** 10**;**  total **=** total **-** k\_2 **\*** 10**;**  k\_3 **=** total **/** 5**;**  total **=** total **-** k\_3 **\*** 5**;**  k\_4 **=** total**;**  printf**(**"need %3d 25\_cent\n"**,**k\_1**);**  printf**(**"need %3d 10\_cent\n"**,**k\_2**);**  printf**(**"need %3d 5\_cent\n"**,**k\_3**);**  printf**(**"need %3d 1\_cent\n"**,**k\_4**);**  **return** 0**;**  **}** |

### 运行结果



## C基础习题

以下各标识符中，哪些是合法的用户标识符？

|  |  |
| --- | --- |
| （1）main | （11）$100 |
| （2）MAIN | （12）I am |
| （3）a205 | （13）max(5) |
| （4）\_exp | （14）begin |
| （5）a\_b | （15）#define |
| （6）3.5 | （16）NONAMEOO |
| （7）A[10] | （17）proc |
| （8）A.name | （18）and |
| （9）%d | （19）not |
| （10）float | （20）WHile |

注：字母或下划开始的后跟若于字母或下划线或数字组成的序列；不能和保留字相同；避免和系统函数名相同；大小写字母有别！

2．选择

（1）18/4\*sqrt(4.0)/8的值的类型？（B）

A．int B．float C．char D．不确定

（2）在C中，一个unsined int型数据的表示范围是（D）

A．0-127 B．0-225 C．0-32767 D．0-65535

（3）(-15)%(-8)的值是（B）

A．-7 B．7 C．1.875 D．非法

3．下列常量哪一组全是合法的？（1）

（1）288，-079，3.4e-2，’A’，10111

（2）-0，1e14，.5678，’ABC’，oxabc

（3）+1，3.e-3，’4’，ox5a，.0

（4）15，16.8e+3，069，101，oxabcd

（5）2L，345e8，’\n’，’\\’，’\101’

4．下列常量中哪一组都是非法的？（ ）

（1）0a，e1.5，ox7，’.’，158

（2）088，ox9afg，65538，3E24，’\089’

（3）32768，1.oe584，3.5e-476，”A”，9FBA

（4）-32769，45678L，12e-2.1，OX9ab，’\063’

（5）9abH，177Q，07777，-9999.999，’\t’

5．计算下列表达式

（1）15+30%4 17

（2）1.5+15/2 8.5

（3）10==9+1 1

（4）x=10，3+8，y=20 赋值语句与空语句

（5）x=y=500 赋值语句

（6）设x=1，y=2，求x++，y++ 两个变量的数值不变

（7）5>3 1

（8）10<500 1

（9）5>3&&5>8 0

（10）10>5||5<10 1

# 实验总结

随着时间的推移，我的编译器选择历经轮转，从最初的古老的TC2.0，到Code::Blocks集成开发环境，再到Visual Studio 2017，最终还是回到了GNU平台上来，使用开源的一套库进行实验。在此期间，云南大学也从一个普通的211大学跻身双一流大学行列，高级语言程序设计这门课程是否也该升级一下？TC2.0这个编译器，界面十分古朴，在几十年前绝对算是一流的软件，但是现在，确实落后了，这主要是因为它无法编译在64位系统下运行的程序。但是TC2.0有很多的优势，比如完全可视化的编译过程，不会生成很多附加文件，这一点Visual Studio就太过专业化。经过对《UNIX环境高级编程》[1]这本书的学习，还有诸如*Harley Hahn's Guide to Unix and Linux*[2]这本书的阅读，我觉得基于Shell的UNIX环境似乎是最适合新手学习的。

本次实验，集中主要精力，在以前版本的基础上，对文档结构进行了重整，看起来自然了很多，目录也规范了很多。有关编程的规范性问题，参考林锐高质量C/C++编程指南的第一版[3]。

# 参考文献

1. Stevens, W.R. and S.A. Rago, *UNIX环境高级编程*. 2nd ed. 2005, 北京: 人民邮电出版社.

2. Hahn, H., *Harley Hahn's Guide to Unix and Linux*. 2009, New York: McGraw-Hill.

3. **林锐**, *高质量 C++/C 编程指南*. 1.0 ed. 2001.

# 教师评语