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：选择结构程序设计 | | |
| 实验编号：NO.2 | 实验日期：2018年8月7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. 进一步掌握运行一个C语言程序的方法和步骤。
2. 分清C语言的符号、标识符、保留字的区别。
3. 掌握C语言的数据类型，会定义整型、实型、字符型变量以及对它们的赋值方法。
4. 学会数据输入方式和数据输出格式及各种格式转意符。
5. 学会使用C的运算符以及用这些运算符组成的表达式，特别是自加（++）和自减（--）运算符的使用。

# 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

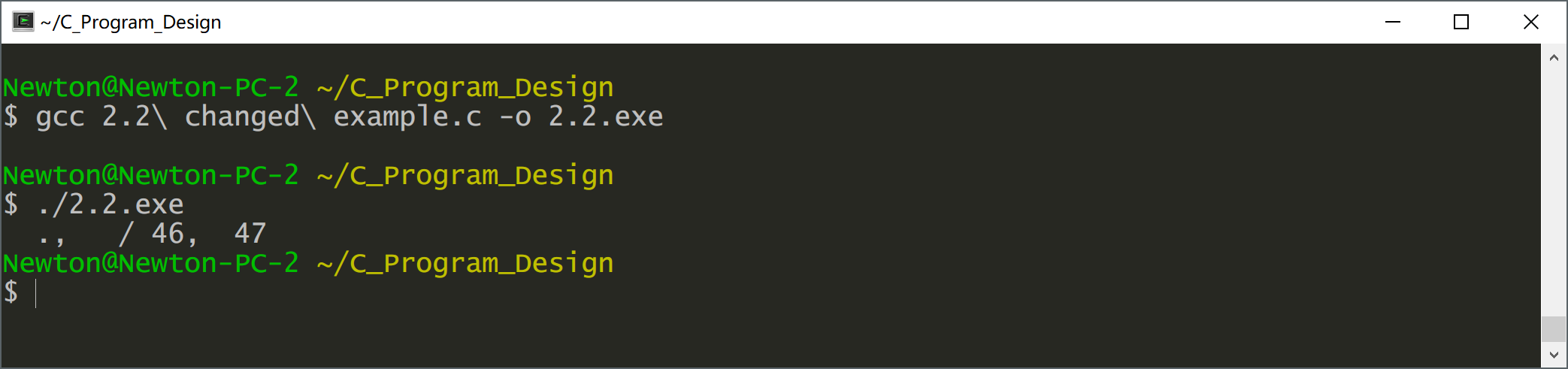
## 题1

输入并运行下面程序，分析其运行结果。这是一段对变量进行不同格式输出打印的程序，在第二次实验里，变量的类型被替换。

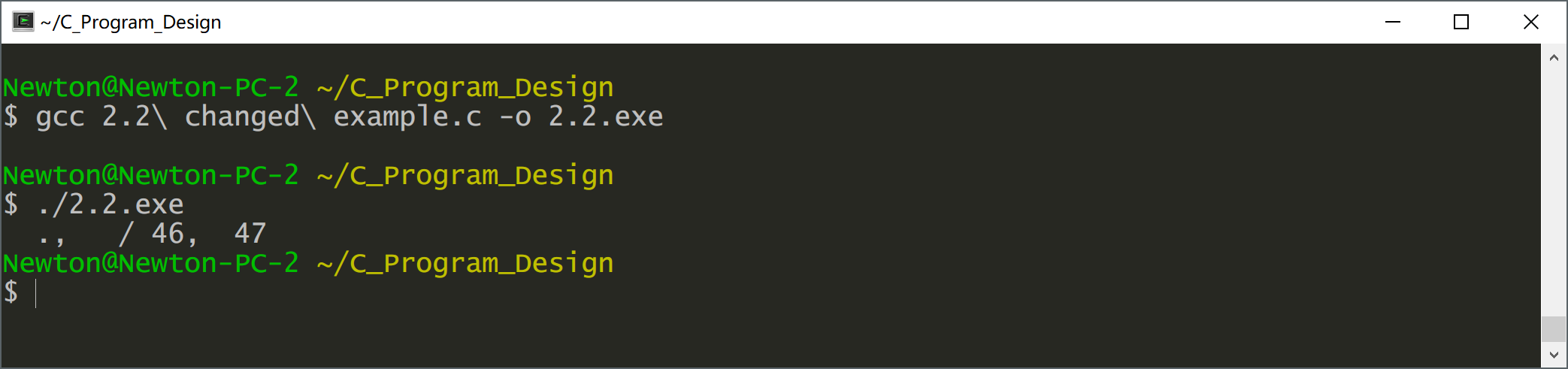
### 程序代码

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | /\*  \* filename: 2.1 example.c  \* property: example  \*/  #include <stdio.h>  int main**()**  **{**  char c1**,** c2**;**  c1 **=** 46**;**  c2 **=** 47**;**  printf**(**"%3c, %3c"**,** c1**,** c2**);**  printf**(**"%3d, %3d"**,** c1**,** c2**);**  **return** 0**;**  **}** |

### 运行结果



将程序第二行改为：int c1**,**c2;再运行，分析其结果。



注：实际本例体现出C语言的一种特性（灵活），整型变量与字符型变量可以相互转换。

## 题2

输入并运行下面程序。运行这个程序，分析结果，特别注意输出c1，c2的值是什么？什么原因？（1）将输入e和f、u和v的语句分别改为：

scanf**(**"%d %d"**,&**e**, &**f**);**

scanf**(**"%d %d"**,&**u**, &**v**);**

运行并分析结果。最后，将程序的第一行加命令：

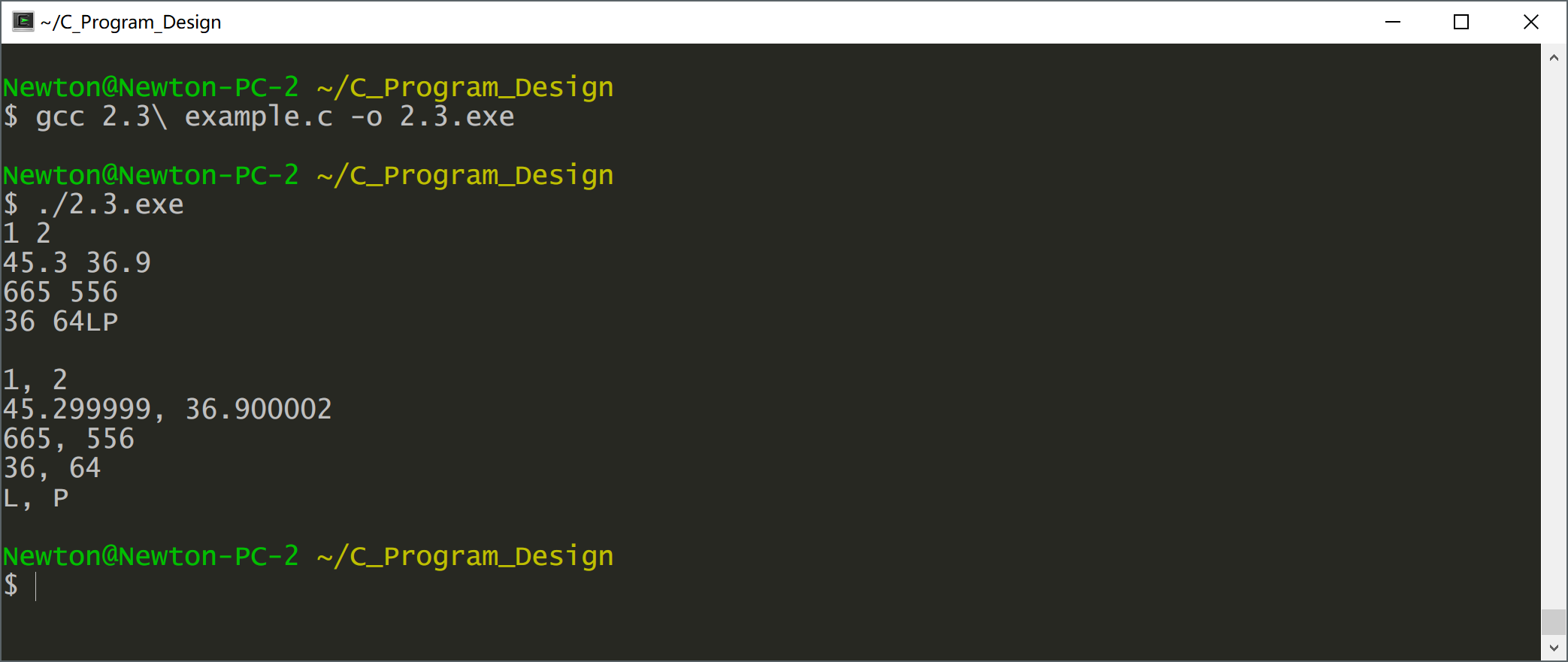
#include <math.h>

运行并分析结果。

### 程序代码

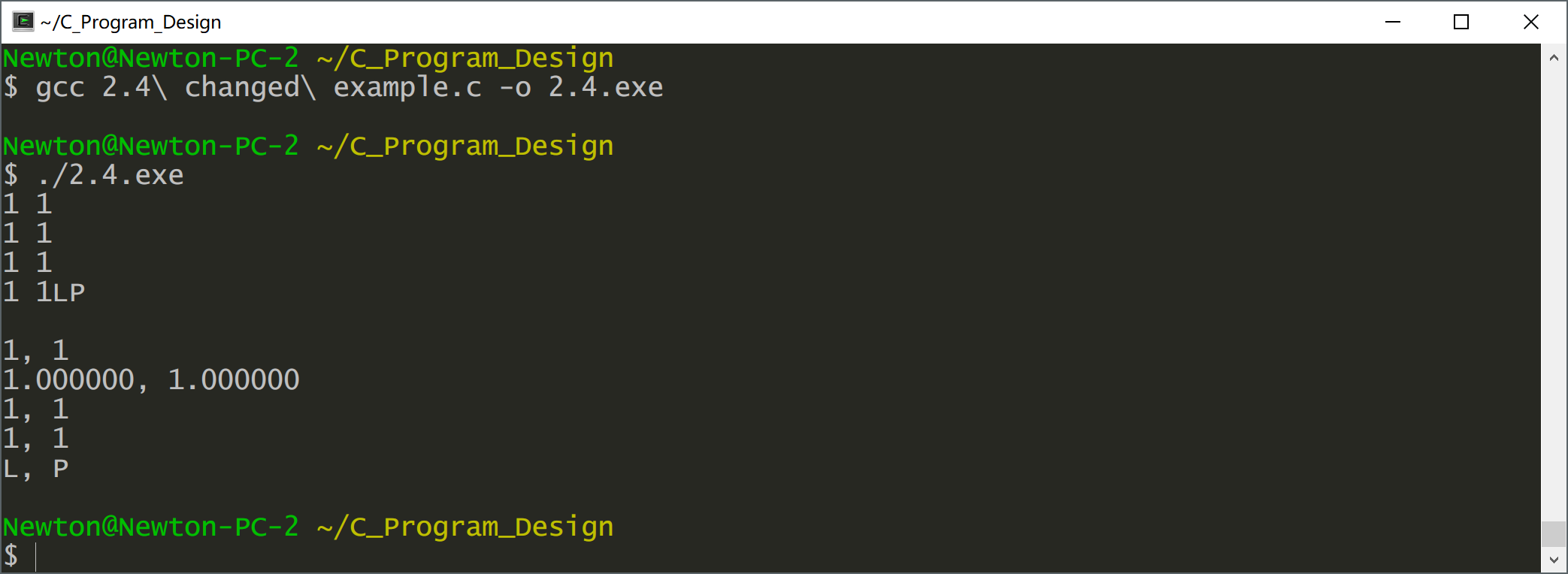
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | /\*  \* filename: 2.3 example.c  \* property: example  \*/  #include<stdio.h>  int main**()** **{**  int a**,** b**;**  float c**,** d**;**  long e**,** f**;**  unsigned int u**,** v**;**  char c1**,** c2**;**  scanf**(**"%d %d"**,&**a**,** **&**b**);**  scanf**(**"%f %f"**,&**c**,** **&**d**);**  scanf**(**"%ld %ld"**,&**e**,** **&**f**);**  scanf**(**"%o %o"**,&**u**,** **&**v**);**  scanf**(**"%c %c"**,&**c1**,** **&**c2**);**  printf**(**"\n"**);**  printf**(**"%d, %d\n"**,**a**,** b**);**  printf**(**"%f, %f\n"**,**c**,** d**);**  printf**(**"%ld, %ld\n"**,**e**,** f**);**  printf**(**"%o, %o\n"**,**u**,** v**);**  printf**(**"%c, %c\n"**,**c1**,** c2**);**  **return** 0**;**  **}** |

### 运行结果

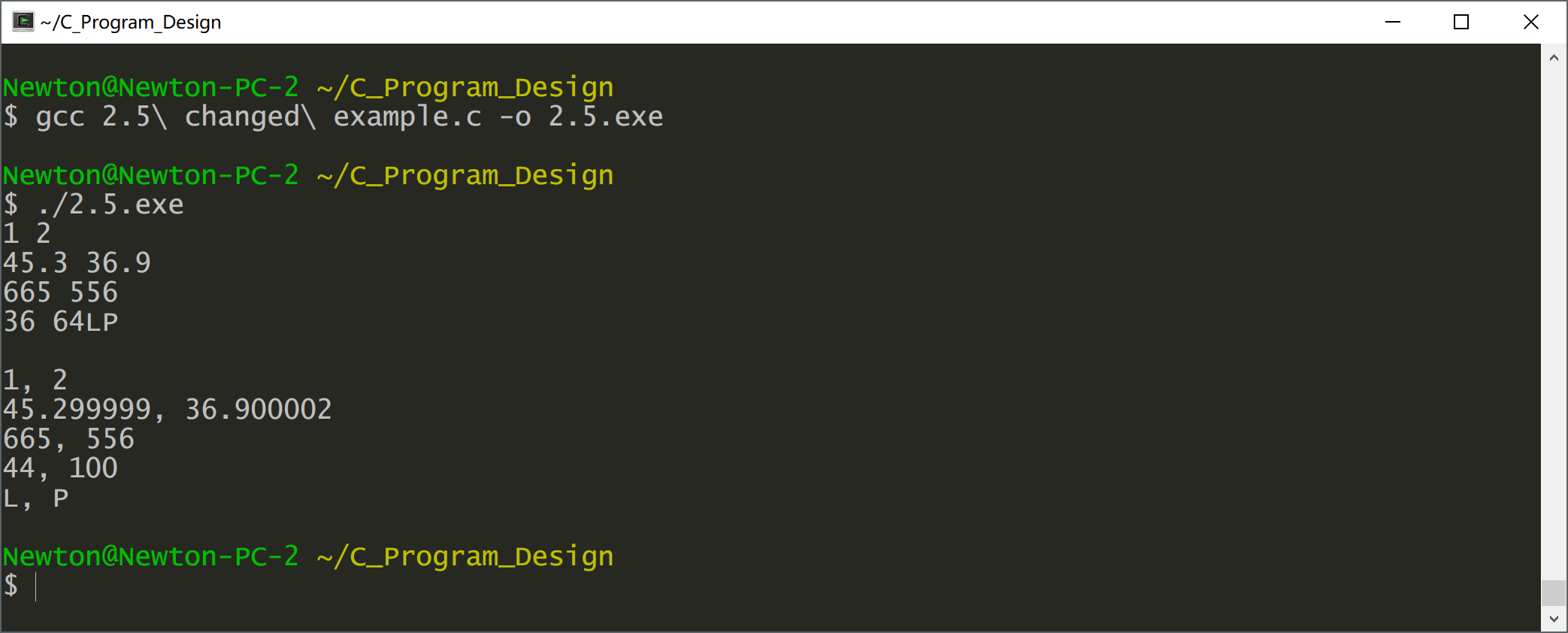


在输入数字的时候，数字彼此之间可以用空格分割，但是在输入字符的时候，由于空格与回车也算做字符，所以要想输入英文字母之类的，应该不加空格地直接输入。

更改输入方式之后，运行结果如下



添加math库之后，运行结果如下：



可以发现，在这个例子之下，添加数学库之后，运行结果没有发生任何变化。

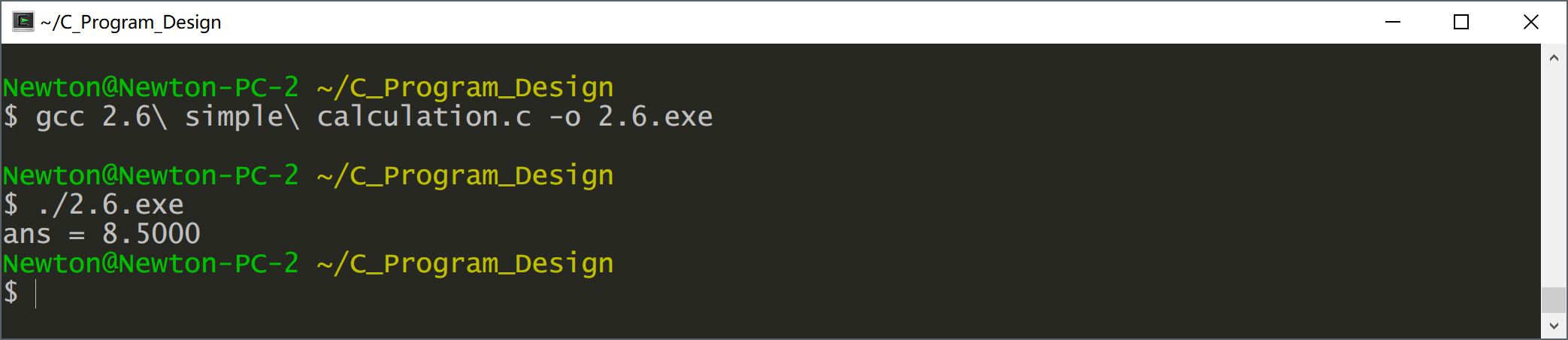
## 题3

编写一个程序，求表达式x **–** z **%** 2 **\* (** x **+** y**) %** 2 **/** 2的值。设，，。

### 程序代码

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | /\*  \* filename: 2.6 simple calculation.c  \* property: homework  \*/  #include<stdio.h>  int main**()** **{**  float x **=** 8.5**;**  float y **=** 2.5**;**  float z **=** 4.0**;**    printf**(**"ans = %2.4f"**,** x**-(**int**)((**int**)**z **%** 2 **\*** **(**double**)(**int**)(**x**+**y**))** **%** 2 **/** 2**);**  **return** 0**;**  **}** |

### 运行结果



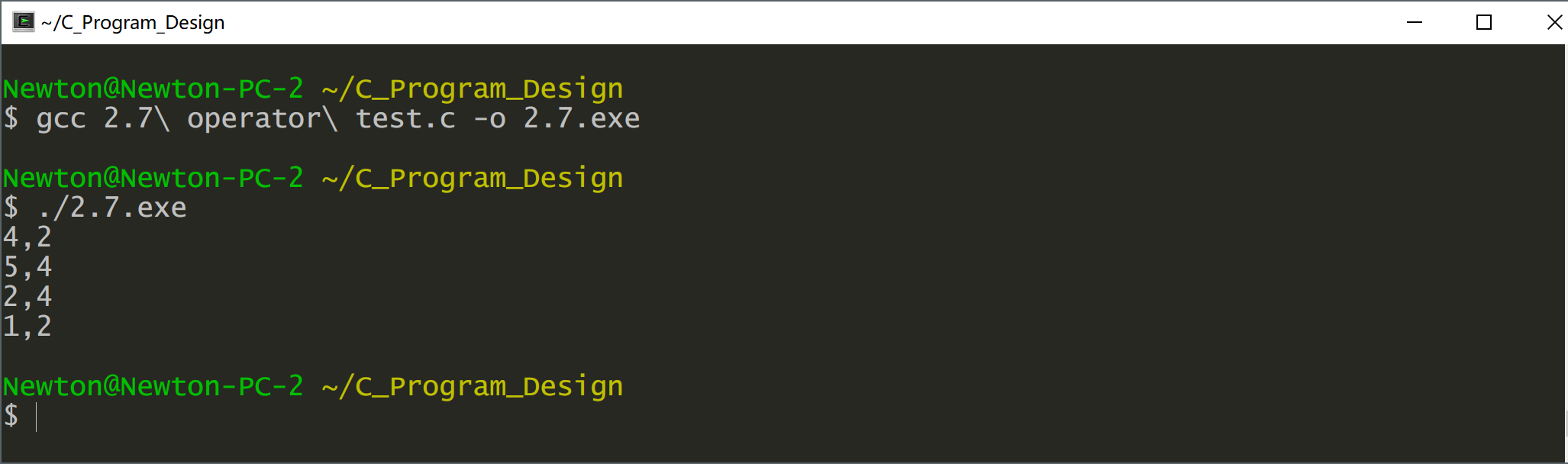
## 题4

先分析下面程序的结果，然后再上机运行，看结果上否一致。

### 程序代码

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | /\*  \* filename: 2.7 operator test.c  \* property: example  \*/  #include <stdio.h>  int main**()** **{**  int x**,** y**,** z**;**  x **=** y **=** z **=** 3**;**  y **=** x**++** **-**1**;** printf**(**"%d,%d\n"**,** x**,** y**);**  y **=** **++**x **-**1**;** printf**(**"%d,%d\n"**,** x**,** y**);**  y **=** z**--** **+**1**;** printf**(**"%d,%d\n"**,** z**,** y**);**  y **=** **--**z **+**1**;** printf**(**"%d,%d\n"**,** z**,** y**);**  **return** 0**;**  **}** |

### 运行结果



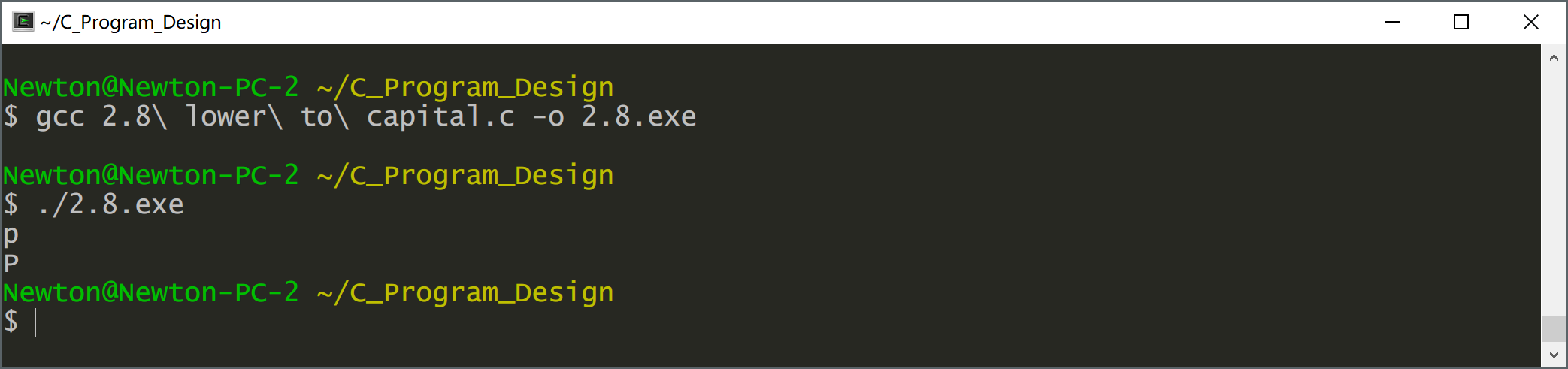
注：本例学生注意，自增自减运算符，先赋值后自增（自减）和先自增（自减）后赋值的问题。

编写一个程序，将输入的小写字母改写成大写字母并输出。提示：可采用getchar()函数输入字符，并利用for循环语句。当然也可用其它方法，只要能实现其功能即可。下面给出一个语句段，学生补充一个完整的程序后，上机进行调试。

### 程序代码

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | /\*  \* filename: 2.8 lower to capital.c  \* property: example  \*/  #include<stdio.h>  int main**()** **{**  char c1**,** c2**;**  c1 **=** getchar**();**  c2 **=** c1 **-** 32**;**  printf**(**"%c"**,** c2**);**  **return** 0**;**  **}** |

### 运行结果



## 习题[1]

1．（判错）下述论断哪些是不对的？

（1）每个C语言程序有且仅有一个主函数main()。

（2）Ｃ语言程序的每一行都用分号结尾。

（3）Ｃ程序的执行从第一行开始到最后一行结束。

（4）Ｃ程序的每一行只能写一条语句。

（5）Ｃ程序的一条语句可以占多行。

（6）一个Ｃ程序可有一个或多个函数组成。

（7）在Ｃ程序中，注释说明只能写在一条语句的末尾。

（8）在一个Ｃ程序中，主函数必须放在程序的首部。

（9）在一个Ｃ程序中，主函数main()可以放在程序的任何位置上。

（10）在Ｃ程序中，注释部分是用花括号括起来的。

## 程序查错

### （1）

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | #include <stdio.h>  #include <math.h>  int main**()** **{**  float r**,** s**;**  s**=** π **\*** r **\*** r**;**  printf**(**"s = %f\n"**,** s**)**  **}** |

答：不是ASCII字符集里的可用文字。

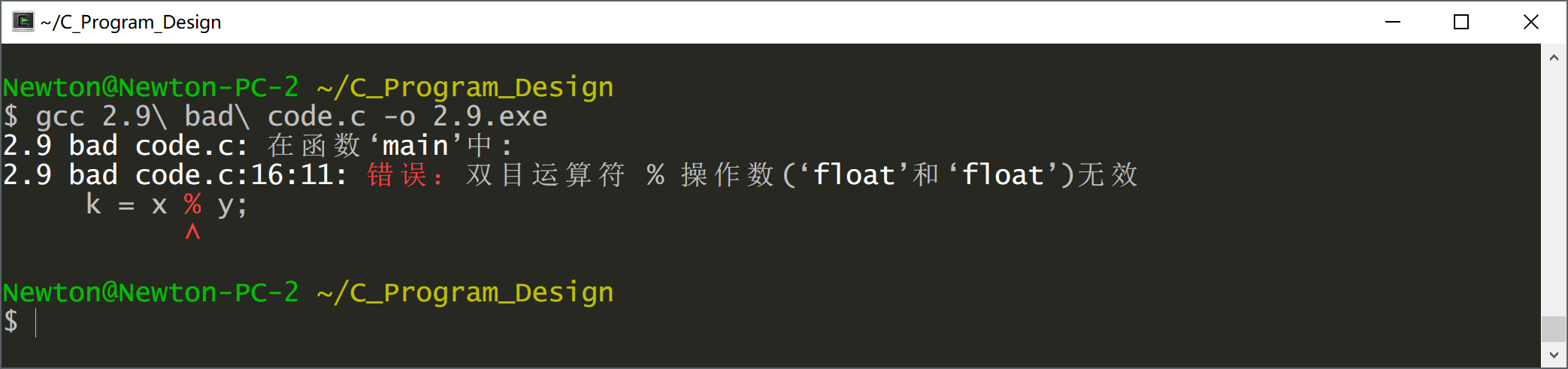
### （2）

运行看看结果如何？为什么？

#### 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 | /\*  \* filename: 2.9 bad code.c  \* property: example, this code could not run.  \*/  #include<stdio.h>  int main**()** **{**  int i**,** j**,** k**;**  float x**,** y**,** z**;**  scanf**(**"%d %f %f"**,&**i**,** **&**j**,** **&**k**);**  scanf**(**"%d %f %f"**,&**x**,** **&**y**,** **&**z**);**    i **=** i **+** x**;**  y **=** y **+** j**;**  z **=** i **+** j**;**  k **=** x **%** y**;**    printf**(**"%d, %f, %f\n"**,**i**,** j**,** k**);**  printf**(**"%f, %d, %d\n"**,**x**,** y**,** z**);**  **return** 0**;**  **}** |

#### 运行结果



编译器首先会进行报错，第16行的求余数运算在浮点型数据上面不能运算。之后有很多警告，不希望对数据进行输入或者输出的时候不按照定义的类型来。因为类型与使用不匹配，所以导致warning与error。

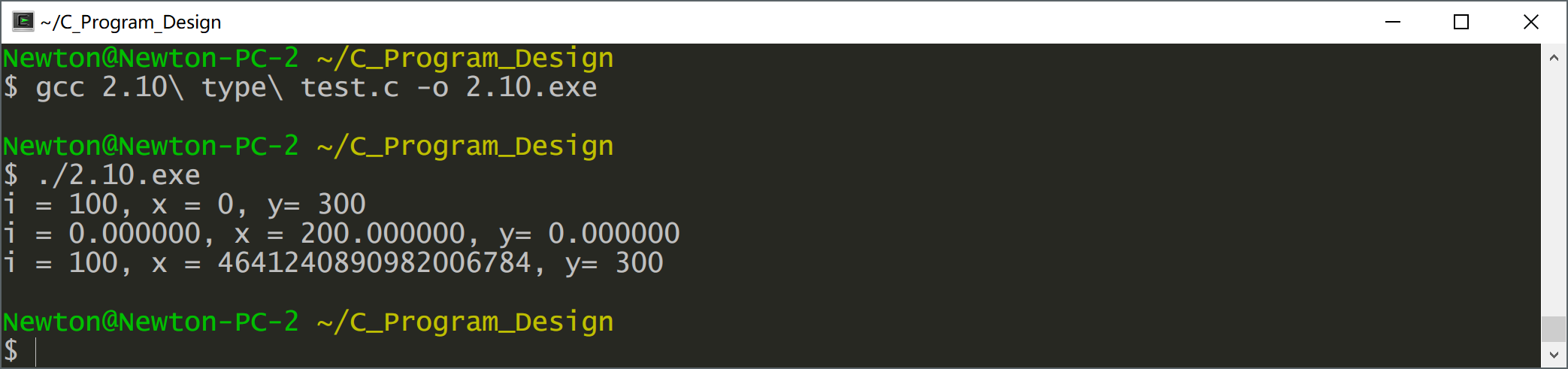
### （3）

运行试试看，结果如何？为什么？

#### 程序代码

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | /\*  \* filename: 2.10 type test.c  \* property: example, this code could not run.  \*/  #include<stdio.h>  int main**()** **{**  int i **=** 100**;**  float x **=** 200**;**  long y **=** 300**;**  printf**(**"i = %d, x = %d, y = %d\n"**,** i**,** x**,** y**);**  printf**(**"i = %f, x = %f, y = %f\n"**,** i**,** x**,** y**);**  printf**(**"i = %ld, x = %ld, y= %ld\n"**,** i**,** x**,** y**);**  **return** 0**;**  **}** |

#### 运行结果



浮点型数据与整型数据的存储方式不一样，强行进行打印，就会出现内存读取与设想的不一样。打印和运算不一样，运算会进行一些类型转换，但是打印则不同。

## 题5

参考下面程序，如何改写输入函数，并配合正确的键盘输入方法才能使x，y和ch分别获得值10，100和’A’？ 若将输入语句改为scanf**(**"%d,%d,%c"**,&**x**,&**y**,&**ch**);**或者scanf**(**"%d,%c,%d"**,&**x**,&**ch**,&**y**);**或者scanf**(**"%c,%d,%d"**,&**ch**,&**x**,&**y**);**结果将会如何？

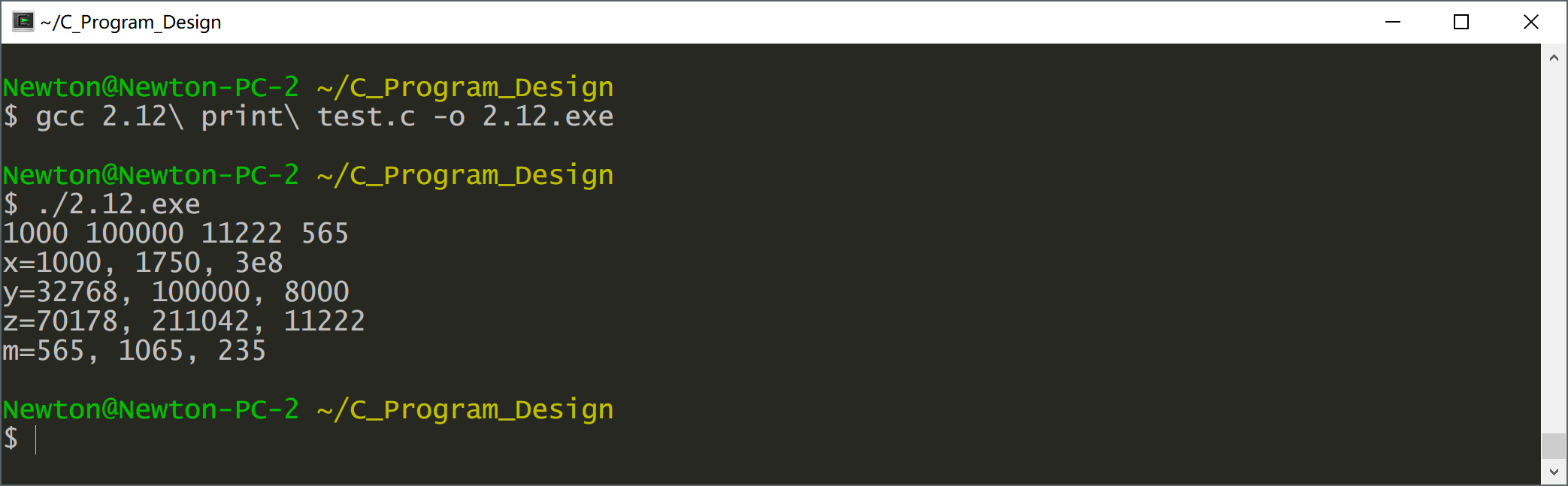
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | /\*  \* filename: 2.11 input test.c  \* property: example  \*/  #include<stdio.h>  int main**()** **{**  int x**,** y**;**  char ch**;**  scanf**(**"x = %d, y = %d, ch = %c"**,** **&**x**,** **&**y**,** **&**ch**);**  printf**(**"x = %d, y = %d, ch = %c\n"**,** x**,** y**,** ch**);**  **return** 0**;**  **}** |

答：键盘输入的方法是在对应位置上输入同样的字符，如果不改变scanf函数引号里的内容，就要输入x=3, y=2, ch=1（其他的数字也好，按自己的需要来。）

下面程序对输入有何要求？利用它可以作什么？

### 程序代码

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | /\*  \* filename: 2.12 print test.c  \* property: example  \*/  #include<stdio.h>  int main**()** **{**  int x**,** y**,** z**;**  long m**;**  scanf**(**"%d %o %x"**,** **&**x**,** **&**y**,** **&**z**);**  scanf**(**"%ld"**,** **&**m**);**  printf**(**"x=%d, %o, %x\n"**,** x**,** x**,** x**);**  printf**(**"y=%d, %o, %x\n"**,** y**,** y**,** y**);**  printf**(**"z=%d, %o, %x\n"**,** z**,** z**,** z**);**  printf**(**"m=%ld, %lo, %lx\n"**,** m**,** m**,** m**);**  **return** 0**;**  **}** |

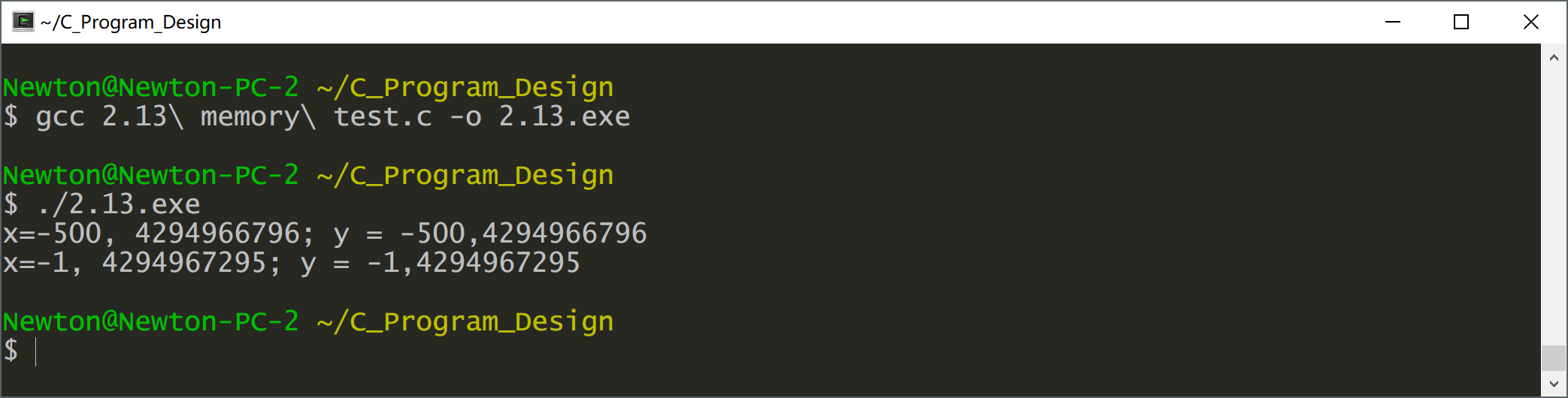


可以通过这种方式进行进位制转化。

## 补码

你记得补码[[1]](#footnote-1)是何意义吗？下面的程序说明了什么？如果将，各赋值为。或者各赋值为和-，输出结果如何？为什么？

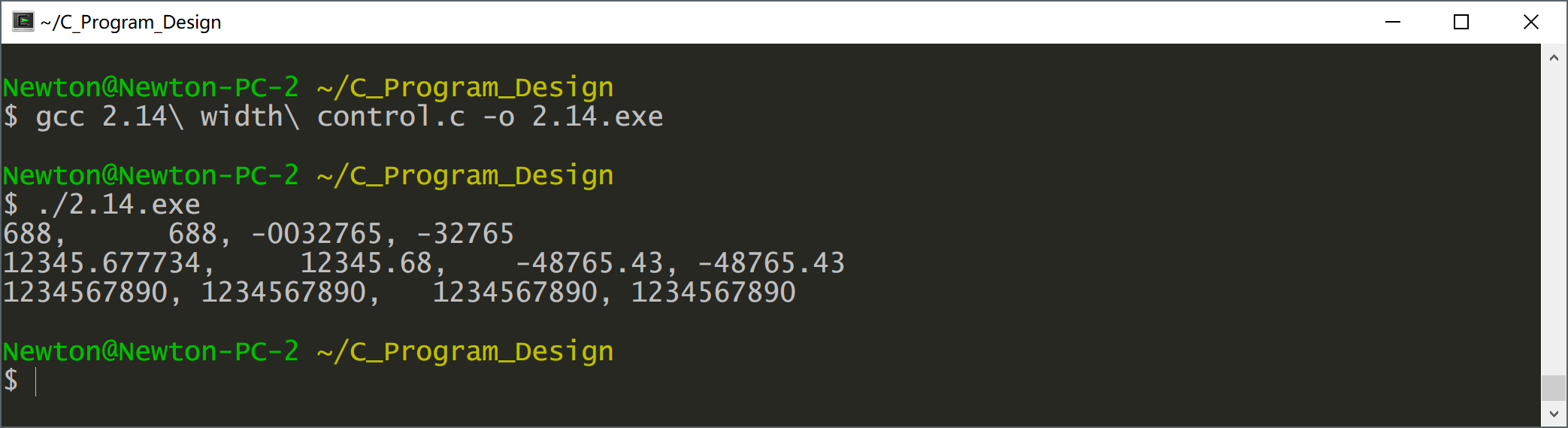
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | /\*  \* filename: 2.13 memory test.c  \* property: example  \*/  #include<stdio.h>  int main**()** **{**  int x **=** **-**500**;**  long y **=** **-**500**;**  printf**(**"x=%d, %u; y = %ld,%u\n"**,** x**,** x**,** y**,** y**);**  x **=** **-**1**;**  y **=** **-**1**;**  printf**(**"x=%d, %u; y = %ld,%u\n"**,** x**,** x**,** y**,** y**);**  **return** 0**;**  **}** |



这个涉及数据的存储形式，如果都改成，那么根据无符号类型的定义，的32位补码是，输出无符号，那么认定之为正数，转换成十进制就是4294967295，即。

## 输出宽度及控制

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | /\*  \* filename: 2.14 width control.c  \* property: example  \*/  #include<stdio.h>  int main**()** **{**  int i**,** j**;**  float x**,** y**;**  long int m**;**  i **=** 688**;**  j **=** **-**32765**;**  x **=** 12345.678**;**  y **=** **-**48765.432**;**  m **=** 1234567890**;**    printf**(**"%d, %8d, %08d, %-8d\n"**,** i**,** i**,** j**,** j**);**  printf**(**"%f, %12.2f, %12.2f, %-12.2f\n"**,** x**,** x**,** y**,** y**);**  printf**(**"%ld, %lu, %12ld, %-12ld\n"**,** m**,** m**,** m**,** m**);**  **return** 0**;**  **}** |



可以发现，%08d中，0的作用就是将不足八位的地方，在高位补上0；而%-8d中负号的作用是改变数据输出的偏向规则。正常是居右显示，加了负号就是往左靠，不足的地方补空格。

# 实验总结

通过调试例题的程序，思考所提出的问题，收获了关于数据的存储、输入以及输出的一些问题的解决方案。

# 参考文献

1. Prata, S., *C Primer Plus 中文版*. 6th ed. C和C++实务精选. 2016, 北京: 人民邮电出版社.

# 教师评语

1. 补码，整数*X*补码指的是：对于正数与原码相同；对于负数，数符位为1，数值位为*X*的绝对值取反后加1，即为反码加1。 [↑](#footnote-ref-1)