
程序设计实训题集

张慧翔编

Version 0.1

2018 年 01 月 01 日

目录

介绍.....	3
1、Number 类.....	4
题目 1 二进制与十进制转换器 (id: number1)	4
题目 2 类似找零钱的操作 (id: number2)	4
题目 3 根据两地坐标, 计算两地之间的距离 (id: number3)	5
题目 4 计算 Fibonacci 序列的值 (id: number4)	6
题目 5 输出既是回文又是素数的值的阶乘 (id: number5)	7
题目 6 输出 9*9 乘法表 (id: number6)	7
题目 7 打印输出日历 (id: number7)	7
2、Text 类.....	8
题目 1 判断一个字符串是不是回文 (id: text1)	8
题目 2 摩斯码生成器 (id: text2)	9
题目 3 重建 grep 命令工具 (id: text3)	9
题目 4 获取出现频率最高的 10 个单词 (id: text4)	10
题目 5 C#代码注释删除 (id: text5)	10
3、Classes 类.....	12
题目 1 实现电影租借的功能 (id: class1)	12
题目 2 计算图形面积及周长 (id: class2)	13
4、Web 类.....	13
题目 1 屏幕抓取 (id: web1)	13
5、DataBase 类.....	13
题目 1 实现数据库的操作 (id: database1)	13
6、NetWork 类.....	14
题目 1 端口扫描工具 (id: network1)	14
题目 2 获取当前天气情况 (id: network2)	14
7、XML 类.....	14
题目 1 实现 XML 文件的生成与解析 (id: xml1)	14

介绍

以个人为单位上交是实验代码文件，个人文件夹以“班号_学号”形式命名；文件夹内包含实训报告、练习题源代码。实训报告电子版采用 pdf 格式，以“report.pdf”形式命名；每项练习的源代码只允许单个文件，以“题目 id.py”形式命名。

参考目录结构如下：

09061701_2014201325

| report.pdf

| number1.py

| number2.py

.....

1、Number 类

题目 1 二进制与十进制转换器 (id: number1)

描述：代码实现用户输入二进制数据，输出其对应的十进制的值，或者用户输入十进制，输出其对应二进制的值。提示：使用一些简单的二进制及十进制进行测试，确保输出结果正确。例如使用那些很容易进行计算的值 3,4,5。尝试讲一个数字（以整除的方式）除以 2，并且记录其余数。注意， $N\%2$ 的余数总是 1 或 0。将除 2 得到的结果再次除 2，直到不能被 2 整除为止。增大难度：将数据转化为八进制或者其它进制。

输入：十进制数和二进制数，以英文逗号分开，如：101,101011

输出：（1）十进制对应的二进制值，二进制对应的十进制值，中间以英文逗号分开，如：1100101,43；（2）当输入格式不正确时，输出：“格式错误”

示例：

输入

101,101011

输出

1100101,43

输入

asd,sd

输出

“格式错误”

代码方法定义：

```
def d2b(decimal_int):    return
def b2d(binary_string)
```

题目 2 类似找零钱的操作 (id: number2)

描述：编写程序，该程序显示用户物品列表及其金额，然后请用户输入要购买的物品及其支付金额。该程序实现找给用户找零的操作，最大面值为 100 元。找寻的零钱有以下几种：50 元，20 元，10 元，5 元，1 元，5 角，1 角。比如，物品：12.3 元，支付 100 元，程序应找寻：1 个 50 元，1 个 20 元，1 个 10 元，一个 5 元，2 个 1 元，1 个 0.5 元，2 个 0.1 元。提示：首先计算出差额，然后用整除的方式计算。商品列表及对应金额如下：

"item01": 2.3,

"item02": 35.8,

```
"item03": 16.3,  
"item04": 12,  
"item05": 13.6,  
"item06": 29,  
"item07": 17.4,  
"item08": 63.9,  
"item09": 56.7,  
"item10": 23.8,
```

输入：要购买的商品及支付的金额，以英文逗号分开，如：item01,5

输出：（1）当商品存在且支付金额大于等于商品金额时，输出各个面值的数目及对应个数，如：50*0,20*0,10*0,5*0,1*2,0.5*1,0.1*2；（2）无此商品时，输出“无此商品,请重新选择...”；（3）支付金额不足时，输出：“支付金额不足,请重新支付...”

示例：

输入：item01,5

输出：50*0,20*0,10*0,5*0,1*2,0.5*1,0.1*2

输入：item00,8

输出：无此商品,请重新选择...

输入：item02,6

输出：支付金额不足,请重新支付...

代码方法定义：

```
def get_changes(goods_pay):  
def list_goods():
```

题目 3 根据两地坐标，计算两地之间的距离（id：number3）

描述：用户以十进制形式输入两地经纬度的坐标，程序计算两地之间的距离，并以千米为计量单位的形式进行输出。提示：地球并不是一个平面。为简单起见，不考虑椭圆形的影响，因此我们假定地球是圆的。第一个帮助解决这个问题的是 WiKi 百科上的 Haversine 公式，这个公式可以计算两个点之间的距离，其输入参数包括两点的经纬度，地球的半径（6371km）。使用你知道的点进行测试，这将有助于你发现你程序是否能够输出正确结果。提高难度：使用余弦定理来进行举

例的计算，这样精确度就能下降到 1 米左右。

参考网站：https://en.wikipedia.org/wiki/Haversine_formula

输入：坐标 1 的经度纬度，坐标 2 的经度纬度。中间以英文逗号分隔，如：
112.3,25.3,152.6,36.4

输出：（1）两地之间的距离，以 km 为单位：如 4007.65756018；（2）输入数据格式或者范围不合法时，输出“输入有误”的提示。

示例

输入：56.3,114.5,56.9,23.4

输出：输入有误

输入：112.3,25.3,152.6,36.4

输出：4007.65756018

代码方法定义：

```
def haversine_formula(longitude01, latitude01, longitude02, latitude02):
```

题目 4 计算 Fibonacci 序列的值（id: number4）

描述：编写代码，使用递归和常规循环操作计算 Fibonacci 序列第 N 项的值。用户输入 N，程序输出该序列计算得到的第 N 项的值。提示：递归实现：该题是学习递归实现很好的例子，但是采用递归实现效率很低；常规的循环操作：使用之前两个值的和作为该序列的下一个值，第二种方法因为无需进行多次调用，因此性能较高，本题要求两种方法均实现。增加难度：通过图形的形式说明递归实现为什么比常规的循环要效率低。

输入：N 的值，即 Fibonacci 序列的项数，如：12

输出：（1）采用常规循环和递归实现的 Fibonacci 序列第 N 项的对应的值，中间以英文逗号分隔，如：144,144；（2）输入参数异常时，提示“参数异常”

示例

输入：12

输出：144,144

输入：fibo

输出：参数异常

代码方法定义：

```
def fibonacci_recursion(term):
```

```
def fibonacci_loop(term):
```

题目 5 输出既是回文又是素数的值的阶乘（id: number5）

描述：分别写出判断回文的函数、判断素数的函数、实现阶乘的函数，实现对用户输入的十进制值进行判断，如果通过，输出该数字的阶乘。

输入：将要判断的十进制的数字，如：11

输出：（1）如果该数字即是回文又是素数，则输出其阶乘，中间以英文逗号分隔；如 11,39916800；（2）如果该数字非回文或者非素数，则输出“非回文或者非素数”；（3）如果输入非十进制，则输出“参数异常”

示例：

输入：11

输出：11,39916800

输入：100

输出：非回文或者非素数

输入：dsfd

输出：参数异常

代码方法定义：

```
def is_palin(num):  
def is_prime(num):  
def get_num_factorial(num):
```

题目 6 输出 9*9 乘法表（id: number6）

描述：编写程序输出九九乘法表。

输出格式：

```
1 * 1  = 1  
1 * 2  = 2    2 * 2  = 4  
1 * 3  = 3    2 * 3  = 6    3 * 3  = 9  
...  

```

题目 7 打印输出日历（id: number7）

描述：输入年份和月份，输出对应的日历。

输入：年份和月份，中间以英文逗号分隔，如：2012,11

输出：（1）对应的日历；（2）输入年份（1900-2100）或者月份（1-12）区间异常时，输出“日期有误”；（3）输入参数异常时，输出“参数异常”。

示例：

输入: 2013,15

输出: 日期有误

输入: calendar,7

输出: 参数异常

输入: 2012,11

Mon	Tue	Wed	Thu	Fri	Sat	Sun
			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	28	29	30		

输出:

代码方法定义:

```
# 判断年份是否为闰年
```

```
def is_leap_year(year):
```

```
# 获取一个月份的天数
```

```
def get_num_of_days_in_month(year, month):
```

```
# 获取从 1800 年 1 月 1 日到现在一共有多少天
```

```
def get_total_num_of_day(year, month):
```

```
# 返回该月的第一天从该周的第几天开始
```

```
def get_start_day(year, month):
```

2、Text 类

题目 1 判断一个字符串是不是回文 (id: text1)

描述: 回文即是无论顺着读还是倒着读, 都一样。比如 “race car” .编写代码判断输入字符串是不是回文。如果是, 打印 “The string '_____’ is a palindrome” 。提示: 简单方法, 将字符串反转, 然后与先前的进行对比。增加难度: 如果你还没做过此类题, 自己编写反转方法。不要使用语言自带的反转方法。

输入: 判断是否为回文的字符串, 如 “race car”

输出: (1) 是回文时, 输出 “The string ' race car ' is a palindrome” ; (2) 不是回文时, 输出 “The string ' race car ' is not a palindrome”

示例:

adbdcgs

输出:

The string 'adbdcgs' is not a palindrome

输入:

abdc dba

输出:

The string 'abdc dba' is a palindrome

题目 2 摩斯码生成器 (id: text2)

描述: 编写代码实现从用户输入一段字符串, 输出其对应的摩斯码 (@代表点, #代表破折号)。提示: 该题主要是将字符转换为其对应的摩斯电码。字符串可以被看作是一个字符序列, 因此从开始的位置开始循环每一个字符, 并且同时从摩斯码表中找到其对应的摩斯码。这种方式下, 你可以迅速将字母与其对应的摩斯码进行对应。提高难度: 将摩斯码反转为原始的字符串, 这需要你确定哪些空格是原始的, 那些是转化生成的。

参考网站: https://en.wikipedia.org/wiki/Morse_code

输入: 字符串, 将要转换为 Morse 码的字符串;

输出: 转化为 Morse 码之后的字符串

示例

输入

i am morse 258

输出

@@ -@# ## -## ### @#@ @@@ @ -@@###@ @ @ @###@ @

题目 3 重建 grep 命令工具 (id: text3)

描述: 在 Linux/Unix 系统中, grep 是一个长久支持的命令工具, 创建你自己的 grep 工具。提示: 此题是关于正则表达式的。从简单的开始, 逐步深入。如果最简单的测试允许查找一个单词或者短语, 甚至是一段文件列表, 然后打印包含这个单词或者短语的所有行的行数。提高难度: 查看你所创建的这个工具, 能支持多少标志和格式。你能否指定输出管道? (打印至文件或者屏幕)

输入: 文件的文件名, 以及即将要查询的单词, 中间以英文逗号分隔, 如:
grep.txt,session

输出: 匹配该字符串所在的文本文件的行数

示例:

输入:

grep.txt,session

输出:

8,69,70,90,96,97,98,99,144,157,

题目 4 获取出现频率最高的 10 个单词 (id: text4)

描述: 在数据分析中, 提取热度最高的但此时经常要进行的操作, 请编写代码, 实现从 TXT 文本文件中提取出现频率最高的前十个单词。提示: 逐行读取文本内容, 并进行切分, 并逐个统计该行单词的数目信息, 存储于字典中, 最终对字典中的数据进行排序, 可转化为列表之后排序, 并输出前 10 个出现频率最高的单词及其出现的次数。提高难度: 查看你所创建的这个工具, 能支持多少标志和格式。你能否指定输出管道? (打印至文件或者屏幕)

输入: 文本文件的文件名, 如: **GoneWithTheWind.txt**

输出: 前 10 个高频词汇及其出现的次数, 单词及出现次数以英文冒号分隔, 不同单词信息之间以空格分隔。

示例:

输入:

GoneWithTheWind.txt

输出:

the:376 and:258 to:191 of:188 a:131 in:119 was:83 had:81 she:72 that:69

题目 5 C#代码注释删除 (id: text5)

描述: 编写代码实现将 C#代码文件中的注释进行删除。

提示: C#代码中包含了三种注释符:

(1).单行注释 `//`

(2).多行注释 `/* 要注释的内容 */` (注意:多行注释不能嵌套使用)

(3).文档注释 `///` (文档注释可以用来注释方法,也可以用来注释类)

思路: 首先, 读取源代码, 读取过程中, 跳过单行注释和文档注释所在行, 将剩余的形成为字符串;

然后, 使用正则表达式, 删除字符串中的多行注释内容。

最后, 将剩余的内容写入新的文件中。

参考网站: https://en.wikipedia.org/wiki/Morse_code

示例:

输入：

GoneWithTheWind.txt

内容如下：

```
class Program
{
    static void Main(string[] args)
    {
        //这一行代码的作用是将 hello world 打印到控制台中
        Console.WriteLine("Hello world");
        // Console.WriteLine("不想输出的代码");    此行代码被注销了
        //这行代码的作用是将程序暂停在这个地方.
        Console.ReadKey();
    }
}

class Program
{
    static void Main(string[] args)
    {
        /*以下是不想要的代码
        Console.WriteLine("Hello world");
        Console.WriteLine("Hello world");
        Console.WriteLine("Hello world");
        */
        Console.ReadKey();
    }
}

class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Hello world");
        Console.ReadKey();
    }
    /// <summary>
    /// 这个方法的作用就是求两个整数之间的最大值
    /// </summary>
    /// <param name="n1">第一个整数</param>
    /// <param name="n2">第二个整数</param>
    /// <returns>返回比较大的那个数字</returns>
    public static int GetMax(int n1, int n2)
    {
        return n1 > n2 ? n1 : n2;
    }
}
```

输出：

```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Hello world");
        Console.ReadKey();
    }
}

class Program
```

```
{
static void Main(string[] args)
{

    Console.ReadKey();
}
}
class Program
{
static void Main(string[] args)
{
    Console.WriteLine("Hello world");
    Console.ReadKey();
}
public static int GetMax(int n1, int n2)
{
    return n1 > n2 ? n1 : n2;
}
}
```

3、Classes 类

题目 1 实现电影租借的功能（id: class1）

描述：创建一个电影商店是最基本的编程学习方式之一。编写代码实现用户创建自己的电影清单，允许用户向清单中添加自己的电影及关于该电影的信息，每一个电影均有售价，这取决于电影是经典的还是最新发布的，这两者还和 Blu-Ray 的不同。经典电影售价：\$2.25，最新发布的电影\$4.95,此外，如果这些电影还是 Blu-Ray 的，则加价\$1.50。自行设定一个归还日期，如果逾期未归还，则每超过 1 天，多支付\$1.00。提示：电影是一个类，商店又是一个类。商店管理电影对象，如添加电影对象、删除电影对象等，如何设计程序取决于你自己。你可以使用 ArrayList 或者 Vector 或者其他常用的列表创建一个电影列表。一旦电影被租出，则其不能够再次被借,并返回其应当归还的日期。当被归还时，对比当前时间与归还时间，如果逾期，计算应当额外支付多少钱，并提示用户应该支付多少钱，用户支付完成之后，将该电影的租出信息删除。提高：创建一个用户账户管理机制，记录用户的欠费信息，如果欠费，则阻止用户继续进行租借，除非用户完成缴纳欠费金额，同时，你也可以为电影类添加电影图片。你也可以创建管理类用于统计信息，如库存产品的价值以及该商店中的每部电影被拷贝的数目。

题目 2 计算图形面积及周长 (id: class2)

描述：创建长方形类、三角形类、圆形类，这三个类都基于一个 **Shape** 的基类。这三个类中各自至少包含两个方法，一个叫做：**area()** ,返回该图形的面积大小；另一个叫做 **perimeter()** ,返回图形的周长。提示：**Shape** 基类含有两个方法：**area()**, **perimeter()**，但是它们都为空，没有具体实现代码。确保子类都继承该基类，并且根据自身特性覆写该基类的两个方法。

4、Web 类

题目 1 屏幕抓取 (id: web1)

描述：编写代码，允许用户访问某一网页，并且进行网页内指定元素的抓取，比如链接、图片、标签等。提示用户输入标签类型，或者需要匹配的模型，然后访问此页面并返回所有匹配的元素。提示：该项目需要两个参数。第一个参数是想要访问资源的网页地址，通过它我们可以获取一些支持搜索操作的文本。第二个参数是创建的匹配文本的类型的正则表达式，并且通过该正则表达式，返回在网页中匹配的所有元素。如果你对正则表达式不太熟悉，你可以访问诸如 [Regular Expression Tutorial](#) 这种网站，可以帮助你学习一些基础知识，并进行深入学习。注意 **HTML** 源文件包含大量的标签，你可以通过 `<a>` 和 `<div></div>` 来进行数据查询。你可以查询一下“贪婪模式”（换句话说：如 `.*`？）

5、DataBase 类

题目 1 实现数据库的操作 (id: database1)

描述：根据给定的 **TXT** 文件，将其内部数据存储于数据库中，并且实现对数据表的增删改查操作。提示：使用 **python** 自带的 **sqlite3** 数据库进行操作，也可以使用 **mysql** 数据库实现。提高难度：（1）尝试使用 **MySQL** 数据库进行数据的存储与操作；（2）对数据进行更多条件的操作：如排序、正则匹配查询、变量值范围限定；（3）通过 **GUI** 实现以上操作。

数据格式如下：



user_db.txt

6、NetWork 类

题目 1 端口扫描工具 (id: network1)

描述：开发程序允许用户设定端口范围，随后逐个进行测试，验证该端口是否开启。最终将可以使用的端口进行打印输出。提示：首先应该确定用户输入的端口范围合法，一旦端口范围合法之后，使用 `socket` 进行该端口的连接，如果端口允许连接，则其是开启的；如果不允许，则该端口正在被某一应用程序使用或者被防火墙等一些应用阻止。提高：尝试一下看能不能找到各个端口到底被哪些应用程序在使用。

题目 2 获取当前天气情况 (id: network2)

描述：编写程序，用户输入地址，如北京、西安或者邮编，并返回当前该地区的天气情况。将当前温度的最高值和最低值、风速、相对湿度输出显示。提示：首先应该找到可以获取天气信息的开放 API，确定该 API 需要什么样的参数，这样你才能够进行查询并获取返回数据。分析返回数据的格式，因此你才能够进行解析。创建程序用于生成服务请求参数、并合理的显示返回数据。尝试使用国家气象中心的 API。提高难度：获取当前天气的其他信息（晴天、下雨、下雪、大雾），并使用图形的方式显示这些信息，在工程中使用一些动画元素。

7、XML 类

题目 1 实现 XML 文件的生成与解析 (id: xml1)

描述：（1）根据以下信息，使用：`xml.dom.minidom` 生成 xml 文件

（2）使用：`xml.etree.ElementTree` 解析 XML 文件

根节点：Books

根节点属性值：`organization:"nwpu_lib"` `address:"shaanxi"`

子节点包含内容：

```
book_list = [    {"title": "book01", "author": "author01", "price": 98, "publisher": "ABC"},
                  {"title": "book02", "author": "author02", "price": 125, "publisher": "AS"},
                  {"title": "book03", "author": "author01", "price": 75, "publisher": "ABC"},
                  {"title": "book04", "author": "author04", "price": 89, "publisher": "CF"},
                  {"title": "book05", "author": "author01", "price": 99.9, "publisher": "ABC"},
                  {"title": "book06", "author": "author03", "price": 231, "publisher": "CF"},
                  {"title": "book07", "author": "author05", "price": 327, "publisher": "ABC"}
                ]
```

解析输出：

- (1) 输出根节点的值；
- (2) 输出根节点的属性及对应的值
- (3) 输出根节点包含的所有子节点

代码方法定义：

```
# 根据已知信息生成 XML 文件
def create_xml(source_file,xml_name):
# 根据文件名，解析 XML 文件
def parse_xml(xml_file):
```