

苏州大学实验报告

院、系	计算机学院	年级专业	计算机科学	姓名	柯骅	学号	2027405033
课程名称	Python 程序设计					成绩	
指导教师	李正华	同组实验者	无	实验日期			

实验名称 实验九 文件操作

一、实验目的

通过本次实验要达到如下目的：

1. 掌握文件和文件操作的基本概念
2. 掌握文本文件的常用操作和常用方法，并理解相应的编程应用过程
3. 掌握二进制文件的常用操作和常用方法，并理解相应的编程应用过程

二、实验内容

说明：

- 所有题目中涉及到的文件，其内容都是英文字符（没有中文字符）
- 所有涉及的文件在下列地址中下载：

http://hlt.suda.edu.cn/zwfan/python_teach/python-ch10.zip

1. 当前路径下有文本文件 `copy.txt`，编写程序实现文本文件的复制功能。（在当前路径下新建一个 `new.txt` 文件,将文本文件 `copy.txt` 的内容复制到 `new.txt` 文件中）
2. 当前路径下有两个文本文件 `cat1.txt` 和 `cat2.txt`，编写程序将 `cat2.txt` 文件中的内容添加到 `cat1.txt` 文件中。
3. 当前路径下有文本文件 `StrInts.txt`，该文件中有一段英文文章，在该文章中存在一些整数（有正有负）。编写程序读取该文件、并提取出其中所有的整数，然后将这些整数中偶数位数字上全部都为奇数数字的整数保存当前路径的 `ResultInts.txt` 文件中去，保存时每行 3 个数、每个数占 8 列、右对齐左补空格。
4. 当前路径下有一个文本文件 `students_data.txt`，该文件中保存了若干个学生的信息，每个学生信息占一行，学生信息从左到右分别为：学号(int 类型)、姓名(字符串类型)、年级(int 类型)，各学生信息以空格分隔。编写程序完成如下功能：
 - a) 读取所有学生信息。
 - b) 输出所有学生信息到屏幕上，要求每个学生信息占一行，学号占 10 列、左对齐，姓名占 15 列、左对齐，年级占 5 列、右对齐。
 - c) 对所有学生根据其学号按照从小到大排序。
 - d) 删除学号小于指定值 `s1` 的所有学生，其中 `s1` 由键盘输入。
5. 当前路径下有文本文件 `Numbers.txt`，文件中的每一行都是一个浮点数，编写程序读取出所有的浮点数。要求：
 - a) 从小到大排序，将排序后的结果写到当前路径下新生成的一个文本文件 `Sort.txt` 中，每个数占一行。
 - b) 求出这些数字的均值、方差，将结果写到当前路径下新生成的一个文本文件 `Sort.txt` 中，每个数占一行。




- c) 要求生成的文本文件 **Sort.txt** 中同时包含排序和均值、方差的结果。
- 当前路径下有一个文件夹 **Folder**，文件夹下有多个文本文件 **file1~file4**（文件名称和文件内容都是英文的），将这些文本文件内容合并生成一个新的文本文件 **merge.txt** 存放在 **Folder** 文件夹中，不破坏原始文件。
 - 当前路径下有文本文件 **word.txt** 中包含了 20 个英文单词，编写一个程序，删除文件中所有不以元音开头的单词。结果保存在当前路径下新生成的 **new_word.txt** 中。
 - 当前路径下有一个文本文件 **Names.txt**，包含了按照字典序排序的名字。编写一个程序，当用户自己给定一个名字，按照字典序将其插入到正确的位置。如果这个名字已经存在于文件中，则不要插入。
例如：
Names.txt 文件中有如下文本（每个名字占一行）
Aaron
Cornell
用户输入的待插入文本是：Abbott
则生成的新文件夹 **new_word.txt** 的内容是：
Aaron
Abbott
Cornell
 - (选做题)：有一个 100G 的文件 **largefile.txt**（这个文件目前没有 100G，只是做模拟）。实现一个程序，首先输出 **largefile.txt** 的行数，然后无限循环，每次要求用户键盘输入一个行号，然后立刻输出对应行的文本。由于文件很大，不允许将文件内容全部放到内存中；同时也不允许从头扫描文件，得到对应行的文本，因为这样速度太慢。（提示：用二进制模式打开文件，使用 **tell**, **seek** 等方法）

三、 实验步骤和结果（每一道题必须画流程图）

- 第一题程序如下：

```
import os
os.chdir(r'd:/python-ch10/')#改变当前工作目录
f=open('copy.txt')#打开文件 copy.txt
n=open('new.txt','w')#创建文件 new.txt
s=f.readline()
while s!='':#直到文章的末尾
    n.writelines(s)
    s=f.readline()
f.close()
n.close()
#思路：手动读取第一行文字，然后用 while 循环控制复制结束
```

运行结果如下：

 copy.txt	2016/12/13 9:45	文本文档	1 KB
 names.txt	2016/12/13 10:12	文本文档	1 KB
 new.txt	2020/12/29 13:51	文本文档	1 KB

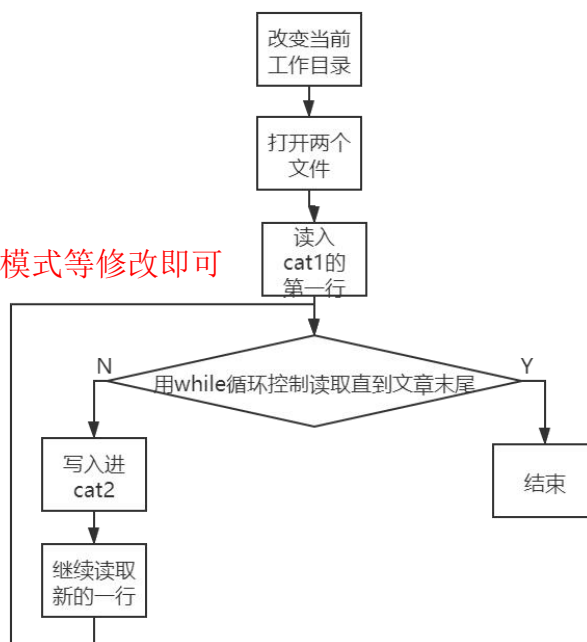
new.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

Parsing is an important problem in natural language processing which has been studied extensively for decades. Between the two basic paradigms of parsing, constituency parsing, the subject of this paper, has in general proved to be the more difficult than dependency parsing, both in terms of accuracy and the run time of parsing algorithms.

2. 第二题程序如下:

```
import os
os.chdir(r'd:/python-ch10/')#改变当前工作目录
f=open('cat1.txt','a')#打开文件 cat1.txt
n=open('cat2.txt')#打开文件 cat2.txt
s=n.readline()
while s!='':#直到文章的末尾
    f.writelines(s)
    s=n.readline()
f.close()
n.close()
#思路: 只需要在第一题的基础上将操作模式等修改即可
```



运行结果如下:

cat2.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

What is the Python Software Foundation?

The Python Software Foundation is an independent non-profit organization that holds the copyright on Python versions 2.1 and newer. The PSF's mission is to advance open source technology related to the Python programming language and to publicize the use of Python. The PSF's home page is at <https://www.python.org/psf/>. Donations to the PSF are tax-exempt in the US. If you use Python and find it helpful, please contribute via the PSF donation page.

cat1.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

What is Python?

Python is an interpreted, interactive, object-oriented programming language. It incorporates modules, exceptions, dynamic typing, very high level dynamic data types, and classes. Python combines remarkable power with very clear syntax. It has interfaces to many system calls and libraries, as well as to various window systems, and is extensible in C or C++. It is also usable as an extension language for applications that need a programmable interface. Finally, Python is portable: it runs on many Unix variants, on the Mac, and on Windows 2000 and later. To find out more, start with The Python Tutorial. The Beginner's Guide to Python links to other introductory tutorials and resources for learning Python.

What is the Python Software Foundation?

The Python Software Foundation is an independent non-profit organization that holds the copyright on Python versions 2.1 and newer. The PSF's mission is to advance open source technology related to the Python programming language and to publicize the use of Python. The PSF's home page is at <https://www.python.org/psf/>. Donations to the PSF are tax-exempt in the US. If you use Python and find it helpful, please contribute via the PSF donation page.

3. 第三题程序如下:

```
#判断是否是符合条件的整数
def func(n):
    s=str(abs(n))
    if len(s)<2:
        return False
    for i in range(len(s)):
        if (i%2==0) and ((ord(s[i])-48)%2==0):
            return False
    return True

#主程序
import os
os.chdir(r'd:/python-ch10/')#改变当前工作目录
f=open('StrInts.txt','rb+')#打开文件
r=open('ResultInts.txt','w+')#创建文件
s=f.readlines()
j,l,n,ss=0,[],-1,' '
for i in range(len(s)):
    ss=bytes.decode(s[i])
    for i in range(len(ss)):
        o=ord(ss[i])
        if o in range(48,58):
            if j<0:
                j=j*10+48-o
            if j>0:
                j=j*10+o-48
            if not(ord(ss[i-1])in range(48,58)):
                j=o-48
            if ss[i-1]=='-':
                j=0-j

        elif j!=0:
            l.append(j)
            j=0

print(l)
ll=[]
for i in l:
    if func(i):
        ll.append(i)
for i in range(((len(ll)-1)//3)+1):#枚举每行
    s=''
    for j in range(3*i,min(len(ll),3*i+3)):
        # 用 min 来防止最后的一两个越界
        s=s+'%8d'%ll[j]
```

```
#s=str.encode(s)
r.writelines(s)
```

```
f.close()
r.close()
```

```
#思路：（1）读入每行
#       （2）寻找数字
#       （3）判断数字
#       （4）输出
```

运行结果如下：

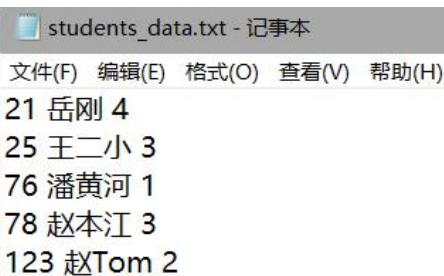


4. 第四题程序如下：

```
import os
os.chdir(r'd:/python-ch10/')#改变当前工作目录
f=open('students_data.txt','rb+')#打开文件
s=f.readlines()
l=[[None for _ in range(3)]for _ in range(len(s))]
for i in range(len(s)):
    ss=bytes.decode(s[i])
    l[i]=list(ss.split())
    l[i][0]=int(l[i][0])
    l[i][2] = int(l[i][2])
    print('%-10d%-15s%5d'%tuple(l[i]))
#排序
l.sort(key=lambda x:x[0])
#清空 txt
f.seek(0)
f.truncate()
#重新输入
s=int(input('please input the s:'))
for i in range(len(l)):
    if l[i][0]>=s:
        ss=' '.join((str(l[i][0]),l[i][1],str(l[i][2]),'\n')).encode()
        print(ss)
        f.write(ss)
#思路：一行一行读取并判断，然后排序，重新键入
```

运行结果如下：

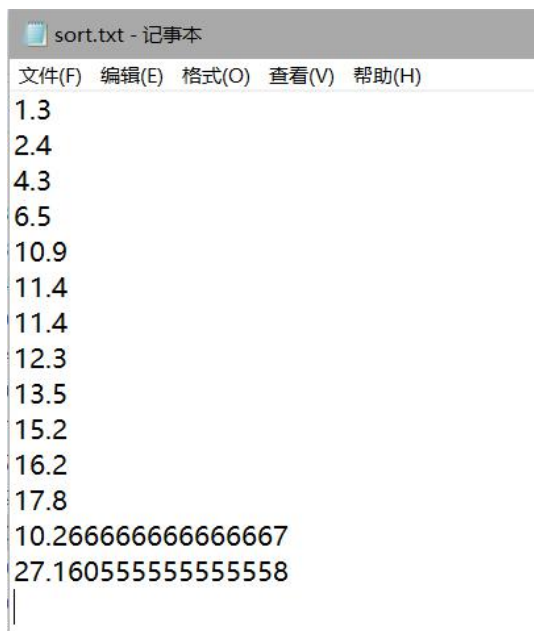
```
25      王二小      3
12      李德华      1
123     赵Tom      2
21      岳刚      4
9       吴志强      2
76      潘黄河      1
78      赵本江      3
2       大沈阳      1
please input the s:20
```



5. 第五题程序如下：

```
import os
os.chdir(r'd:/python-ch10/')#改变当前工作目录
f=open('numbers.txt','rb+')#打开文件
r=open('sort.txt','w+')#创建文件
s=f.readlines()
l=[]
for i in range(len(s)):
    l.append(float(s[i].decode()))
#排序
l.sort()
tot=sum(l)
#平均值
p=tot/len(l)
tot=0
for i in l:
    ss=str(i)+'\n'
    r.write(ss)
    tot+=(i-p)**2
ss=str(p)+'\n'
r.write(ss)
#方差
fang=tot/len(l)
ss=str(fang)+'\n'
r.write(ss)
#思路：整体读入一行行处理，排序，计算平均值和方差，写入文件
```

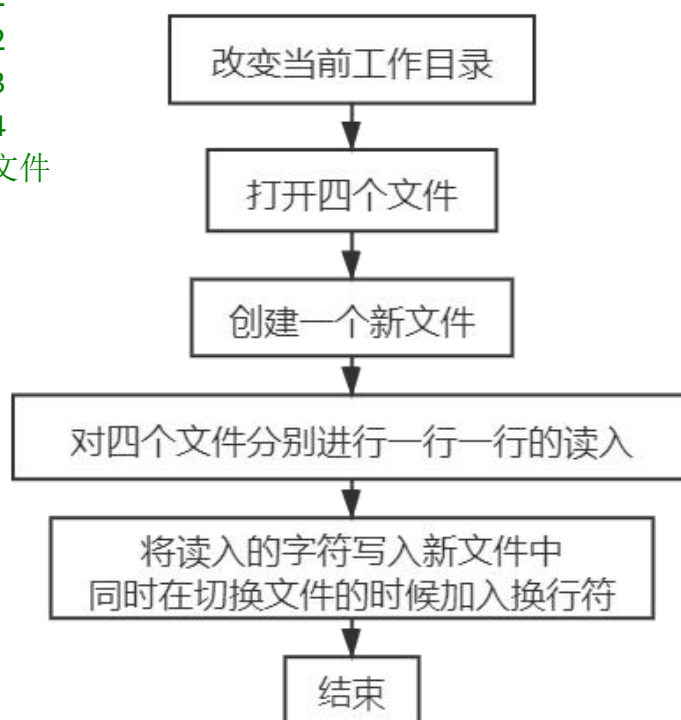
运行结果如下：



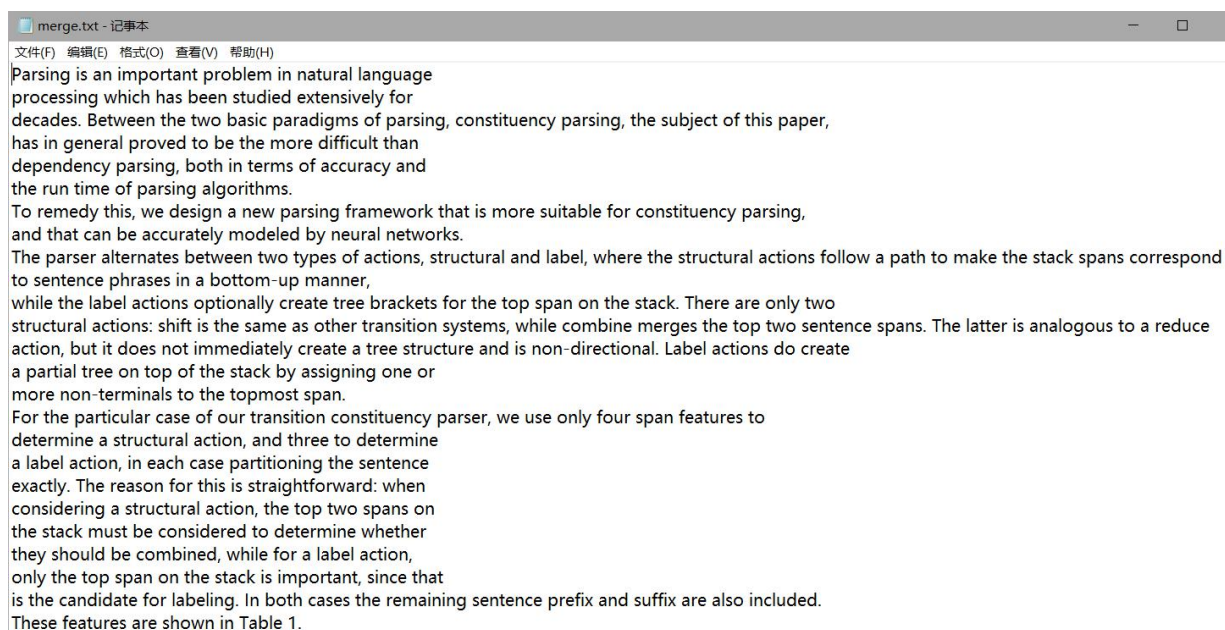
```
sort.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
1.3
2.4
4.3
6.5
10.9
11.4
11.4
12.3
13.5
15.2
16.2
17.8
10.266666666666667
27.160555555555558
```

6. 第六题程序如下：

```
import os
os.chdir(r'd:/python-ch10/folder/')#改变当前工作目录
f1=open('file1.txt')#打开文件 1
f2=open('file2.txt')#打开文件 2
f3=open('file3.txt')#打开文件 3
f4=open('file4.txt')#打开文件 4
n=open('merge.txt','w+')#创建文件
s=f1.readline()
while s!='':#直到文章的末尾
    n.writelines(s)
    s=f1.readline()
n.writelines('\n')
s=f2.readline()
while s!='':#直到文章的末尾
    n.writelines(s)
    s=f2.readline()
n.writelines('\n')
s=f3.readline()
while s!='':#直到文章的末尾
    n.writelines(s)
    s=f3.readline()
n.writelines('\n')
s=f4.readline()
while s!='':#直到文章的末尾
    n.writelines(s)
    s=f4.readline()
#思路：文件一个一个读入，按照第一题的方式进行写入，切换文件中途换行
```



运行结果如下：

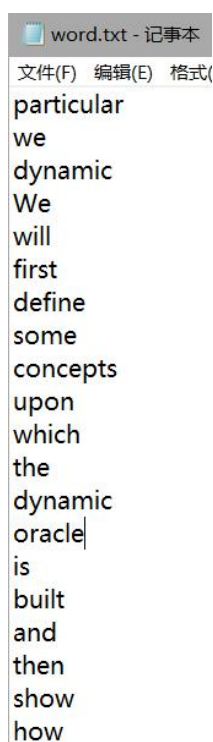


merge.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
Parsing is an important problem in natural language processing which has been studied extensively for decades. Between the two basic paradigms of parsing, constituency parsing, the subject of this paper, has in general proved to be the more difficult than dependency parsing, both in terms of accuracy and the run time of parsing algorithms. To remedy this, we design a new parsing framework that is more suitable for constituency parsing, and that can be accurately modeled by neural networks. The parser alternates between two types of actions, structural and label, where the structural actions follow a path to make the stack spans correspond to sentence phrases in a bottom-up manner, while the label actions optionally create tree brackets for the top span on the stack. There are only two structural actions: shift is the same as other transition systems, while combine merges the top two sentence spans. The latter is analogous to a reduce action, but it does not immediately create a tree structure and is non-directional. Label actions do create a partial tree on top of the stack by assigning one or more non-terminals to the topmost span. For the particular case of our transition constituency parser, we use only four span features to determine a structural action, and three to determine a label action, in each case partitioning the sentence exactly. The reason for this is straightforward: when considering a structural action, the top two spans on the stack must be considered to determine whether they should be combined, while for a label action, only the top span on the stack is important, since that is the candidate for labeling. In both cases the remaining sentence prefix and suffix are also included. These features are shown in Table 1.

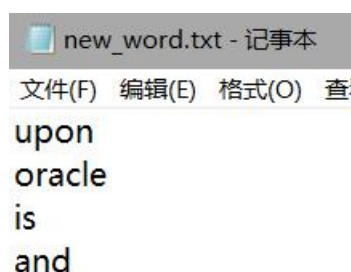
7. 第七题程序如下：

```
import os
os.chdir(r'd:/python-ch10/')#改变当前工作目录
f=open('word.txt')#打开文件
r=open('new_word.txt','w+')#创建文件
s=f.readlines()#储存单词
for i in s:
    if i[0].upper() in ('A','E','I','O','U'):
        r.write(i)#写入
#思路：打开文件，判断是否符合，将符合的写入新文件
```

运行结果如下：



word.txt - 记事本
文件(F) 编辑(E) 格式(O)
particular
we
dynamic
We
will
first
define
some
concepts
upon
which
the
dynamic
oracle
is
built
and
then
show
how

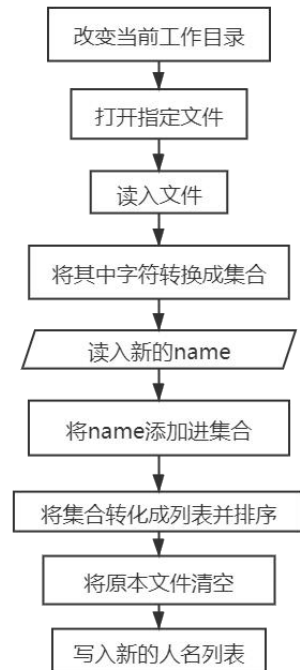


new_word.txt - 记事本
文件(F) 编辑(E) 格式(O) 查
upon
oracle
is
and

8. 第八题程序如下:

```
import os
os.chdir(r'd:/python-ch10/')#改变当前工作目录
f=open('names.txt','r+')#打开文件
s=f.readlines()#储存单词
ss=input('please input a name:')
s=set(s)
s.add(ss+'\n')
s=list(s)
s.sort()
#清空 txt
f.seek(0)
f.truncate()
for i in s:
    f.write(i)
```

#思路: 打开文件, 读入新名字, 用集合去重, 转化成列表排序, 写入



运行结果如下:

please input a name:Coooooooooooooooo

names.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

Aaron
Abbott
Addison
Bard
Bartholomew
Benedict
Coooooooooooooooo
Cornell
Darnell
Edmund
Felix
Gabriel
Harlan
Lennon

9. 选做题:

程序如下:

```
import os
os.chdir(r'd:/python-ch10/')#改变当前工作目录
f=open('bigfile.txt','rb+')#打开文件
l=[]
l.append(0)
s=f.readline()
l.append(f.tell())
while s!=''.encode():#直到文章的末尾
    s=f.readline()
    l.append(f.tell())
print(len(l)-2)#输出行数 (第 0 个和最后一次循环分别多读了一个)
while 1:
    n=int(input('please input the number of the line:'))
    f.seek(l[n-1])
    print(f.readline().decode())
#思路: 用 tell 先扫描一遍文件, 记录每行开头的位置所在, 并记录至一个列表
#      这样在输入之后就可以直接输出
#      已忽略文末空行
```

运行结果如下:

```
840
please input the number of the line:1
.START

please input the number of the line:11
While his case may be extreme, it reflects the vulnerability of many small communities to domineering judges.

please input the number of the line:26
One secretary testified that the judge once called her to his office while wearing nothing at all.

please input the number of the line:
```

四、 实验总结 (包括对老师的建议)

掌握了文件和文件操作的基本概念

掌握了文本文件的常用操作和常用方法, 并理解了相应的编程应用过程

掌握了二进制文件的常用操作和常用方法, 并理解了相应的编程应用过程

对于二进制读取和特定形式读取有了更深层次的认识和理解,

掌握了 flush, eval, open, write, read 等方法的应用, 并能在特定的时间采用合适的方法来解决
问题, 编程能力也得到了很大提升

对于 python 的 os 和 sys 模块有了更深的理解, 同时也进一步认识了工作目录。