

# 综合实验

1. 以下实验任务，请你任意选择其中一个完成（提交截止日期 2015-01-12）。
2. 请仔细阅读实验要求，实现程序功能，并完整地撰写实验报告。
3. 本实验是课程最终的结课实验，每个人独立完成，撰写实验报告，最终成绩将很大程度取决于本实验。
4. 申请课程成绩优秀的同学，需准备大作业的答辩，时间另行通知。答辩时间每人 5 分钟，需要提交纸质实验报告，并做现场程序演示。

计算思维导论 北航计算机学院 实验教学中心

**实验任务 1：**文本处理就是类似 WORD 这类软件的功能，能够对文本的查找、替换、删除等功能，并进行简单意义上的文本分析，如 WORD 能够统计单词数目，进行拼写检查等。

本实验需要你复习本学期学过的数据结构的知识，设计合理的算法，完成以下的实验项目：

- 1) 从互联网上下载一篇你喜欢的英文文章 300 词以上，并去掉标题、上下文空行等，仅保留正文文本，以“.txt”为后缀名保存在本地磁盘。
- 2) 编写一个程序，读入该文件，将每行内容拆解为单词，剥去单词周围的空白字符和标点，并转换为小写。将转换后的结果输出。
- 3) 修改以上程序，统计你的这篇文章中出现的全部单词的总数量。将统计结果输出。
- 4) 继续修改以上程序，统计你的这篇文章里每个单词的使用次数（词频），并将统计结果输出
- 5) 对于纯文本文件，通常认为频率最高的若干个词可以作为文章的关键词，前提是这些词都具有实词意义，而不是英语的助词、语气词（如 a, is, the 等）。实际上你可以用一个停用词表（请你从互联网上搜索一个比较合理的英文停用词表并下载下来供本作业使用），把停用词表中单词排除在外，然后输出你的这篇文章里最经常使用的 10 个具有实词意义的单词（即词频最高的 10 个实词），作为文件的关键词。并将这 10 个词的词频用柱状图（或你认为合理的其它形式的图）加以显示，以便用户获得更直观的可视化结果
- 6) **（选作）**截至到任务（5），你开发的小工具已经具备了不少功能，如果你有兴趣和精力，请把以上你所做的工作用学过的 GUI 的知识包装成一个有图形用户界面的小软件。网上也能找到不少这样的小软件，你可以参考它们的用户界面设计，或者你也可以按照自己的对于 UI 的理解设计一个你自己风格的界面。用你设计的小工具可以实现：（1）显示一个文本文件的内容（2）提供按钮供用户进行总单词数统计、词频统计等（3）能自动生成关键词（词频最高的 10 个实词）

实验指导：

提示 1：请回顾 Python 文件输入输出功能的程序实现

```
try:
    f= #读取文件
    con= #将文件内容读出在变量 con 中
    #关闭文件
except ValueError as ioerror: #处理错误
    print('File already closed {0}'.format(type(ioerror))) #输出错误信息
finally:
    print('operation end') #无论是否异常，都将输出操作结束信息
```

提示 2：在 Python 中，string 模块提供了空白字符串 whitespace，包括空格、制表符、换行符等等，它也提供 punctuation 等，包含了所有的标点字符，另外你也可以考虑用字符串自带的方法 strip\replace\translate 等，方法不唯一。

提示 3：词频统计最为简捷的方法是使用字典。使用下面的语句可以对字典进行排序：

```
wturple=sorted(word_dict.items(), key=lambda d:d[1],reverse=True) #对字典进行排序
```

提示 4：柱状图的显示请参考实验七相关方法进行

## 实验任务 2：24 点纸牌游戏的开发

24 点是一种老少咸宜的游戏，它的具体玩法如下：

给玩家 4 张牌，每张牌的面值在 1~13 之间，允许其中有数值相同的牌。采用加、减、乘、除四则运算，允许中间运算存在小数，并且可以使用括号，但每张牌只能使用一次，尝试构造一个表达式，使其运算结果为 24。

1) 请你根据上述游戏规则构造一个玩 24 点游戏的算法，编程实现。要求如下：

输入：n1, n2, n3, n4

输出：若能得到运算结果为 24，则输出一个对应的运算表达式。

如：输入：11, 8, 3, 5

输出：(11-8) \* (3+5) =24

提示：

算法的设计不唯一，穷举法是最为基本的解法，分治法则会获得比较高一些的效率，请你仔细思考，设计出算法实现该问题并画出算法的流程图。

2) 请为你的 24 点纸牌游戏开发出一个界面

例如：



提示:

1. 你可以百度搜索一下互联网上类似小软件的开发, 试用一下, 确定一个界面风格。
2. 软件主要完成的功能需要以下几个:
  - 1) 提供一个功能能进行随机发牌 4 张 (用纸牌的形状或按钮的形状均可)
  - 2) 提供功能供用户输入关于这 4 张牌的表达式, 并进行计算, 判断结果的正确与否
  - 3) 能提供给用户正确答案

实验指导:

提示 1: 以下给出了穷举法解 24 点的代码框架, 但很显然这种解法并不是最好的求解方法,

你还可以设计其它的算法来解决该问题。

```
def cal(a,b,op):
    if op==0:return(a+b)
    if op==1:return(a-b)
    if op==2:return(a*b)
    if op==3:
        if(b==0.0):
            return(999.0)
        else:return(int(a/b))

def D24(v0,v1,v2,v3):
    op=['+','-','*','/']
    v=[v0,v1,v2,v3]
    count=0
    #四重循环开始穷举四个数字的位置: 4!=24 种

    #三重循环开始穷举三个运算符: 4X4X4=64 种

    #未用循环, 直接穷举三个运算符的优先级: 3!=1=5 种-
    t1=t2=t3=0
    #第 1 种情况 ((a。b)。c)。d 开始计算:
    t1=cal(v[i1],v[i2],f1)
    t2=cal(t1,v[i3],f2)
    t3=cal(t2,v[i4],f3)
    if t3==24:
        print(v[i1],op[f1],v[i2],op[f2],v[i3],op[f3],v[i4])
    count+=1
    #第 2 种情况 (a。b)。 (c。d) 开始计算:

    #第 3 种情况 (a。 (b。c))。d 开始计算:

    #第 4 种情况 a。 ((b。c)。d) 开始计算:

    #第 5 种情况 a。 (b。 (c。d)) 开始计算:

    #穷举结束: 共 24*64*5=7680 种表达式 -----
    if (count==0):
        print("can not calculate 24.",v0,v1,v2,v3);
    else:
        print("has several ways to calculate",count)

v0=int(input("Pls input the 1st number:"))
v1=int(input("Pls input the 2nd number:"))
v2=int(input("Pls input the 3rd number:"))
v3=int(input("Pls input the 4th number:"))
D24(v0,v1,v2,v3)
```

提示 2: 界面设计可参考 GUI 相关知识进行, 用到的控件包括: Button, Entry, Label 等。注意:

内置函数 `eval()` 可以完成把一个字符串作为参数并返回它作为一个 Python 表达式的结果,

例如:

```
>>> eval("2+3*4-5")
```

输出结果为:

9

以上函数可能帮助你获取用户在 Entry 中输入表达式的值

另外, 随机数的生成请参考 Python 的 `random` 模块

计算思维导论 北航计算机学院 教学实验中心