

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 实验六 字符串与正则表达式

一. 实验目的

通过本次实验要达到如下目的：

1. 掌握 Python 字符串相关基本概念
2. 理解并掌握 Python 字符串的各种常用操作及相应的方法
3. 掌握正则表达式的一般应用方法

二. 实验内容

1. 写一个程序，用户输入一个字符串 s，返回一个由 s 的前 2 个字符和后 2 个字符组成的新字符串。如果 s 的长度小于 2，则返回空字符串。

例：输入'python'，返回'pyon'。

2. 写一个程序处理用户输入的字符串，并按用户要求删除其中第 n 个字符，返回删除字符后的字符串。
3. 给定字符串，将其中的单词倒序输出。

例：给定"What a wonderful day!"，输出："day! wonderful a What"。

4. 统计一个字符串中所有字符出现的次数。

例：给定"google.com"，输出：'o': 3, 'g': 2, 'l': 1, 'e': 1, 't': 1, 'm': 1, 'c': 1

5. 英语语法中，动词的第三人称单数形式规则简要概括（不完全）如下：

- a) 如果动词以 y 结尾，则去掉 y 并加上 ies。
- b) 如果动词以 o, ch, s, sh, x, z 结尾，则加上 es。
- c) 默认直接在动词后加上 s。

现在请你写一个程序，对于任意给定的一个动词，返回其第三人称单数形式。

6. 编写一个函数，判断一个密码（用字符串表示）是否是好密码。一个好的密码满足：1) 长度不小于 8；2) 至少含有一个数字；3) 至少含有一个小写字母；4) 至少含有一个大写字母。如果密码是好密码，返回 True，否则返回 False。
7. 编写一个函数，将一个 a 进制的数转换成一个 b 进制的数，其中 a 和 b 都在[2, 16]之间。该函数有 3 个参数，前两个参数分别是 a 和 b，第三个参数是一个字符串，表示 a 进制的数。如果 a 和 b 不在给定范围之内，返回 None，否则返回对应的 b 进制数
8. 一个字符串如果正读和反读都一样，那么它就是一个回文串。编写一个函数，判断一个字符串在下列规则下是否是回文串：1) 忽略所有空格；2) 忽略所有的句号、逗号、感叹号；3) 不区分大小写。如果是回文串，返回 True，否则返回 False。

9. 请利用正则表达式写一个简单的拼写检查程序。实现以下功能：

- a) 两个或两个以上的空格出现时将其压缩为一个。
- b) 在标点符号后加上一个空格，如果这个标点符合之后还有字母。

例：给定字符串："This□□is□□very□funny□and□□□cool.Indeed!"

输出："This□is□very□funny□and□cool.□Indeed!"

其中“□”代表一个空格。

10. 请写一个 Python 程序以尝试解析 XML/HTML 标签。现有如下内容：

```
<composer>Wolfgang Amadeus Mozart</composer>
```

```
<author>Samuel Beckett</author>
```

```
<city>London</city>
```

希望自动格式化重写为：

```
composer: Wolfgang Amadeus Mozart
```

```
author: Samuel Beckett
```

```
city: London
```

三. 实验步骤和结果（每一道题必须画流程图）

1. 第一题程序如下：

```
s=input('please input a string:')
if len(s)<2:
    print()      #s 的长度小于 2
else:
    print(s[:2]+s[len(s)-2:]) #返回一个由 s 的前 2 个字符和后 2 个字符
    组成的新字符串
```

#思路：if 语句结合字符串切片

运行结果如下：

```
please input a string:python  please input a string:a
pyon
```

2. 第二题程序如下：

```
s=input('please input a string:')
n=int(input('please input "n":'))
print(s[:n-1]+s[n:]) #切片
```

#思路：字符串切片

运行结果如下：

```
please input a string:hello
please input "n":2
hllo
```

3. 第三题程序如下:

```
l=list(input('please input a sentence:').split())#split 转变成列表
print(' '.join(l[::-1]))#用 join 拼接
```

#思路: (1) 拆分: split ()
(2) 倒序拼接: 切片+join

运行结果如下:

```
please input a sentence:What a wonderful day!
day! wonderful a What
```

```
please input a sentence:aa bb cc dd asdasdasda
asdadasda dd cc bb aa
```

4. 第四题程序如下:

```
s=input('plsase input a string:')
d={}
for i in s:
    d[i]=d.get(i,0) #防止下一步报错
    d[i]+=1 #value 出现的次数+1
l=list(d.items())
for i in range(len(l)-1):
    print("%s:%d"%(l[i][0],l[i][1]),sep=',end='')
print("%s:%d"%(l[len(l)-1][0],l[len(l)-1][1]),sep=',end='')
```

#思路:
(1) 用字典统计字符串出现次数
(2) 因为输出格式中用逗号间隔,但是最后一个没有逗号,所以转换成列表来判断是否最后一个

运行结果如下:

```
plsase input a string:google.com
'g':2,'o':3,'l':1,'e':1,'.':1,'c':1,'m':1
```

```
plsase input a string:helloween
'h':1,'e':3,'l':2,'o':1,'w':1,'n':1
```

5. 第五题程序如下:

```
s=input('please input a verb:')
l=len(s)
#规则 a)
if s[l-1]=='y':
    print(s[:l-1]+'ies')
#规则 b)
#一行太长, 用反斜杠写到两行
elif s[l-1]=='o' or s[l-1]=='s' or s[l-1]=='x'\
     or s[l-1]=='z' or s[l-2:l]=='ch' or s[l-2:l]=='sh':
    print(s+'es')
#规则 c)
else:
    print(s+'s')
```

#思路: if 语句搭配切片判断并输出

运行结果如下:

```
please input a verb:worry
worries

please input a verb:watch
watches

please input a verb:look
looks
```

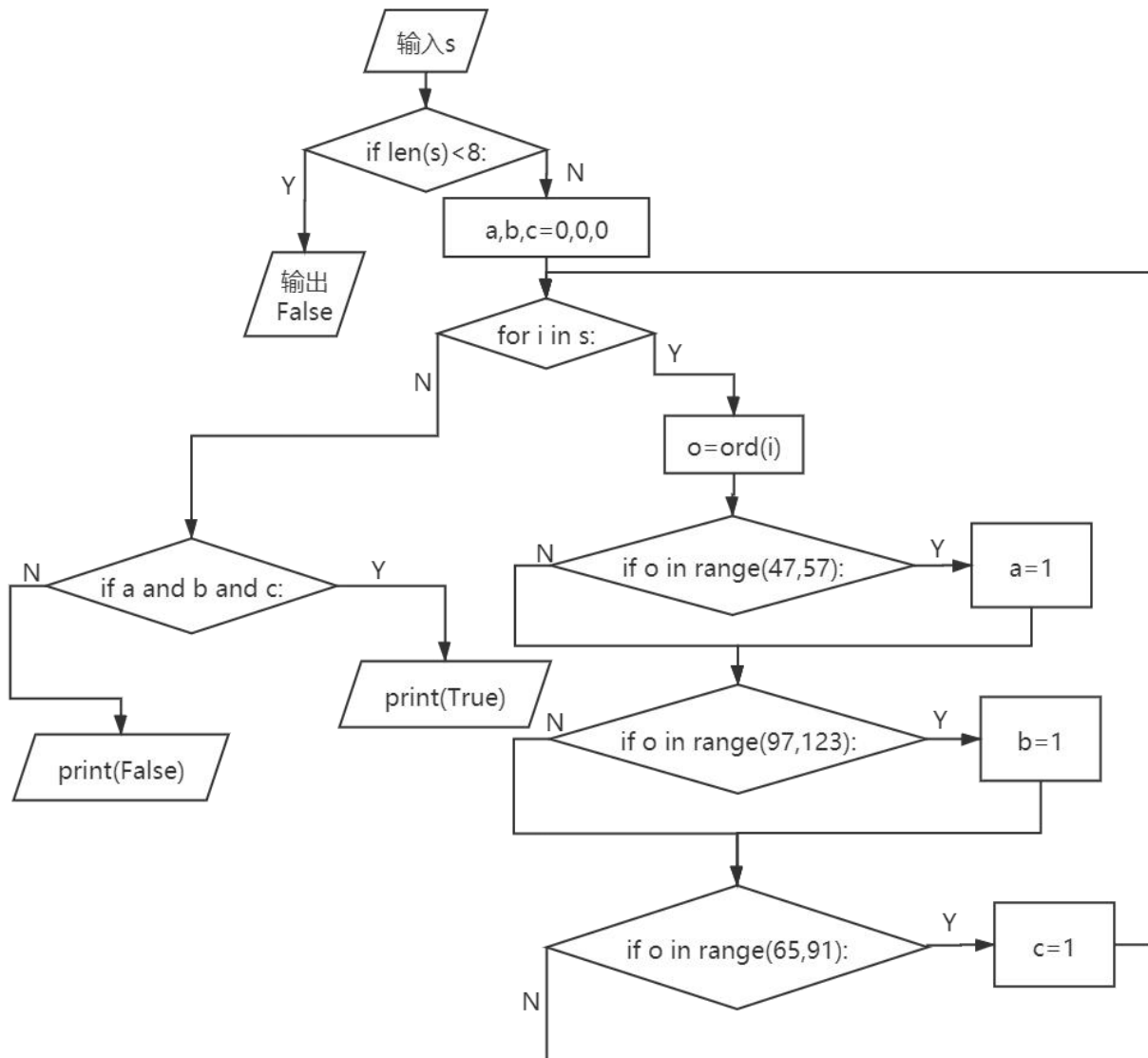
6. 第六题程序如下:

```
s=input('please input a password:')
if len(s)<8: #条件 1)
    print(False)
else: #条件 2) 3)
    a,b,c=0,0,0 # 用 abc 来标记是否符合好密码条件
    for i in s:
        o=ord(i)
        if o in range(47,57):
            a=1
        if o in range(97,123):
            b=1
        if o in range(65,91):
            c=1
    if a and b and c:
        print(True)
    else:
        print(False)
```

#思路:

(1) if 判断条件是否满足

(2) 用标记变量来标记特征



运行结果如下:

please input a password: *qweqw789*
False

please input a password: *ASASqwq8989*
True

7. 第七题程序如下:

```
a=int(input('please input a:'))
b=int(input('please input b:'))
s=input('please input the number:')
#先转化成十进制, 存入 n
n=int(s,a)
#初始化 16 进制以内的字母和数字, 初始化最终输出逆序余数列表 w
aa=[]
while n!=0:
    x=n//b
    y=n%b
    # aa 列表按照规则增加当前余数
    if y in range(10):
        aa.append(y)
    else:
        aa.append(chr(55+y))
    n=x
for i in aa[::-1]: #逆序输出
    print(i,end='')
```

#思路:

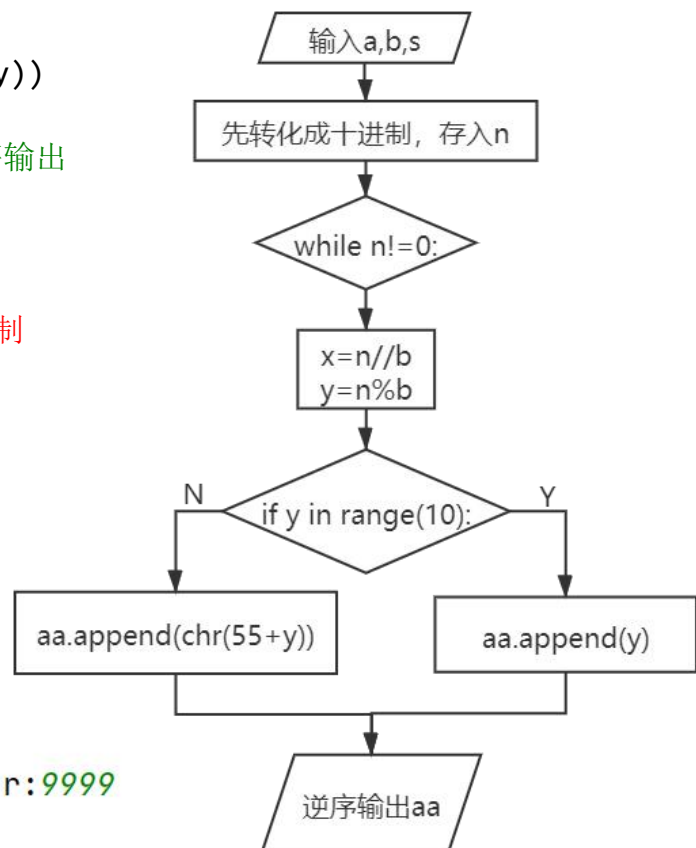
(1) 将 a 进制先转化成 10 进制

(2) 用辗转相除求 b 进制

运行结果如下:

```
please input a:10
please input b:16
please input the number:9999
270F
```

```
please input a:2
please input b:16
please input the number:11111010101
7D5
```



8. 第八题程序如下:

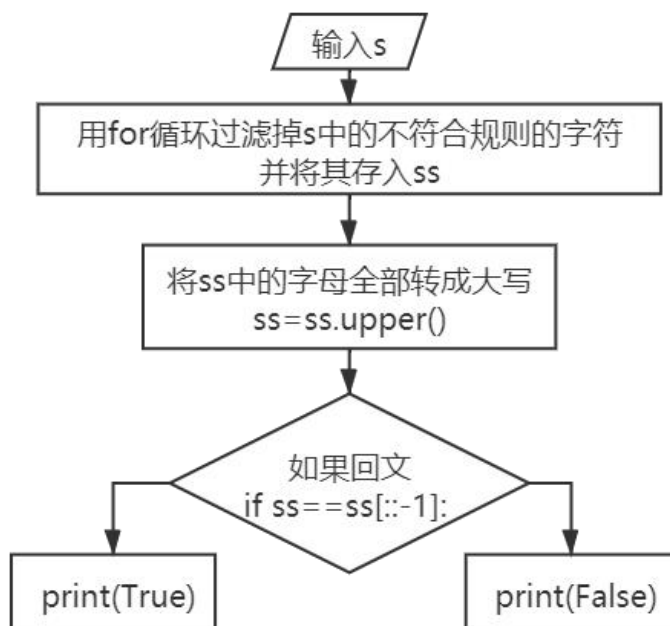
```
s=input('please input a string:')
ss=''
for i in s:
    if not(i in (' ','.',',','!')):    #过滤操作
        ss=ss+i
ss=ss.upper()    #对大小写不敏感
if ss==ss[::-1]:    #回文
    print(True)
else:
    print(False)
```

#思路:

(1) 用 if 语句和 for 循环过滤不需要的东西

(2) 用 upper 对大小写不敏感

(3) 切片判断回文



运行结果如下:

```
please input a string:a..a,.,!!!b bbc    bb    ba.....a
True
```

```
please input a string:a..a,.,!!!b bbc    bb    ba.....aa
False
```

9. 第九题程序如下:

```
import re
s=input('please input a sentence:')
s=re.sub(' +',' ',s)#将多个空格替换成一个空格
s=re.sub('([,?!.;\'\"])(\w+)',r'\1 \2',s)#在标点后加空格
print(s)
```

#思路: 采用正则表达式中的 sub 替换

运行结果如下:

```
please input a sentence:This is very funny and cool.Indeed!
This is very funny and cool. Indeed!
please input a sentence:What a nice day,let's
What a nice day, let' s
```

10. 第十题程序如下:

```
import re
ss=''
for i in range(3):
    s=input()
    s=re.sub(r'<(\w+)>(.+)</\1>',r'\1: \2',s)##re 中的替换
    ss=ss+s+'\n'##多行输出
print(ss)
```

##思路: 类似于第 9 题, 采用 re 中的 sub 替换
用'\n'在多行输出

运行结果如下:

```
<composer>Wolfgang Amadeus Mozart</composer>
<author>Samuel Beckett</author>|
<city>London</city>
composer: Wolfgang Amadeus Mozart
author: Samuel Beckett
city: London
```


拓展题目:

(1) 矩阵乘法

程序如下:

```
n1=int(input('n1:'))
m1=int(input('m1:'))
n2=int(input('n2:'))
m2=int(input('m2:'))
a=[[None for _ in range(m1)]for _ in range(n1)]
b=[[None for _ in range(m2)]for _ in range(n2)]
c=[[0 for _ in range(m2)]for _ in range(n1)]
for i in range(n1):
    for j in range(m1):
        a[i][j]=int(input('A {:0},{:1}:'.format(i,j)))##格式化输出
for i in range(n2):
    for j in range(m2):
        b[i][j]=int(input('B {:0},{:1}:'.format(i,j)))##格式化输出
print(a)
print(b)
for i in range(n1):
    for j in range(m2):
        for k in range(m1):
            c[i][j]=c[i][j]+(a[i][k]*b[k][j])##根据公式

print(c)
```

思路: 根据公式

$$(AB)[i,j] = A[i,1] * B[1,j] + A[i,2] * B[2,j] + \dots + A[i,n] * B[n,j]$$

可以看出采用三重循环可以解决矩阵乘法的问题、

第一第二重枚举 **c** 矩阵的行和列, 第三重枚举公式中的 **i**

运行结果如下:

```
n1:2
m1:3
n2:3
m2:2
A 0,0:1
A 0,1:3
A 0,2:5
A 1,0:4
A 1,1:6
A 1,2:2
B 0,0:2
B 0,1:3
B 1,0:3
B 1,1:2
B 2,0:1
B 2,1:1
[[1, 3, 5], [4, 6, 2]]
[[2, 3], [3, 2], [1, 1]]
[[16, 14], [28, 26]]
```

(2) 快速排序

程序如下:

```
def sort(lst):
    if len(lst)<2: ##无需排序
        return lst
    mid=lst[len(lst) // 2] ##找到基准
    l,r=[],[]
    lst.remove(mid) ##删除基准
    for i in lst:
        if i>=mid: #比基准大放到 r 列表
            r.append(i)
        else:
            l.append(i)
    return sort(l)+[mid]+sort(r) #返回排好序的列表

n=int(input('please input the amount of numbers:'))
a=list(map(int,input('please input the numbers:').split()))
print(sort(a))
```

思路:

- (1) 选择一个元素作为”基准”。
- (2) 数列中除基准外所有元素都和这个基准值进行比较,如果比基准值小就移到基准值的左边,如果比基准值大就移到基准值的右边
- (3) 以基准值左右子列作为新数列,不断重复第一步和第二步,直到所有子集只剩下一个元素为止。

运行结果如下:

```
please input the amount of numbers:9
please input the numbers:5 6 8 7 4 1 2 9 3
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

四. 实验总结 (包括对老师的建议)

掌握了 Python 字符串相关基本概念

理解了并掌握了 Python 字符串的各种常用操作及相应的方法

掌握了正则表达式的一般应用方法

Re.sub(a,b,c)里面有三个参数,不是两个,是返回值不是赋值,需要一个变量名来接收
了解并理解了快速排序的原理

‘.’可以匹配任意字符

贪心就是一直匹配知道匹配到字符串结束 (默认)

不贪心就是遇到匹配成功的字符串就算成功