

Python实验报告

实验六：知识回顾与扩展

姓名：余泓锸 学号：2016326603046

实验目的

确认python实验环境是否完善

对前期上课内容与实验细节进行回顾与复习

解决之前遇到的问题，并加深印象

实验内容

实验题1：

1. 请你编写一个函数，能够判断2个字符串，哪一个更长，输出该字符串和长度

解答：

In [1]:

```
def get_lenstr(a='', b=''):
    la = len(a)
    lb = len(b)
    if la > lb:
        m = a
        lm = la
    else:
        m = b
        lm = lb
    print('最长的字符串为：{0}，其长度为：{1}'.format(m, lm))
```

In [2]:

```
get_lenstr('asfaf', 'sdfsdfsdfs')
get_lenstr()
```

最长的字符串为：sdfsdfsdfs，其长度为：10

最长的字符串为：，其长度为：0

实验题2：

2. 请你编写一个程序，能够解决经典的鸡兔同笼问题，输入头和脚的数量，自动计算鸡和兔的数量，如果解不存在，输出input error；

解答：

In [3]:

```
def jtin_cage(head=0, foot=0):
    if foot%2==1:
        print('The number of feet is odd, and the conditions are not valid! ')
        return False
    num = foot-head*2
    if num<0:
        print('input error')
        return False
    tu = num/2
    ji = head-tu
    if ji<0:
        print('input error')
        return False
    print('鸡有{0}只，兔有{1}只。'.format(ji, tu))
    return True
```

In [4]:

```
while True:
    string = input('请依次输入头和脚的数量(输入quit退出): ')
    if string=='quit':
        break
    try:
        head, foot = map(int, string.split())
    except:
        continue
    jtin_cage(head, foot)
```

请依次输入头和脚的数量(输入quit退出): 5 31
The number of feet is odd, and the conditions are not valid!
请依次输入头和脚的数量(输入quit退出): 4 6
input error
请依次输入头和脚的数量(输入quit退出): 3 32
input error
请依次输入头和脚的数量(输入quit退出): 6 20
鸡有2.0只，兔有4.0只。
请依次输入头和脚的数量(输入quit退出): quit

实验题3：

3.（选做）请你编写一个程序，对输入的4个数字，能够判断这4个数字能否计算24点，可以的话，输出计算方法，不可以，输出none。

解答：

第三题代码一

In [3]:

```
def get_twolist(sz):
    twolist = []
    a = list(sz)
    for i in a:
        b = list(a)
        b.remove(i)
        for j in b:
            twolist.append([i, j])
    return twolist
```

In [4]:

```
def get_relist(sd=[1, 1]):
    relist = []
    oplist = ['-', '+', '*']
    relist.append(sd[0]-sd[1])
    relist.append(sd[0]+sd[1])
    relist.append(sd[0]*sd[1])
    if sd[1]!=0:
        relist.append(sd[0]/sd[1])
        oplist.append('/')
    return (relist, oplist)
```

In [9]:

```
_24(pz=[2, 3, 4, 5]):
    []
stl = get_twolist(pz)
in twolist1:
elist1, oplist1 = get_relist(i)
= list(pz)
.remove(i[0])
.remove(i[1])
or j, jo in zip(relist1, oplist1):
    a.append(j)
    twolist2 = get_twolist(a)
    for k in twolist2:
        relist2, oplist2 = get_relist(k)
        b = list(a)
        b.remove(k[0])
        b.remove(k[1])
        for x, xo in zip(relist2, oplist2):
            b.append(x)
            relist3, oplist3 = get_relist(b)
            for y, yo in zip(relist3, oplist3):
                if abs(y-24.0)<1e-10: # eval函数
                    string = str(i[0])+jo+str(i[1])+'+='+str(j)+' $$ '+str(k[0])+xo+str(k[1])+'+='+str(
                        way.append(string)
                    #return set(way) 解开注释只显示一种方法
            b.remove(x)
        a.remove(j)
    t len(way):
rint('none')
eturn False
n set(way)
```

In [10]:

```
while True:
    string = input('请依次输入四个数(输入quit退出): ')
    if string=='quit':
        break
    try:
        pz = [int(i) for i in string.split()]
    except:
        continue
    way = judge_24(pz)
    if way:
        for i in way:
            print(i)
```

请依次输入四个数(输入quit退出): 2 3 4 5

3+5=8 \$\$ 8-2=6 \$\$ 4*6=24
3+5=8 \$\$ 8+4=12 \$\$ 2*12=24
3-2=1 \$\$ 5+1=6 \$\$ 4*6=24
3+4=7 \$\$ 5+7=12 \$\$ 2*12=24
5+3=8 \$\$ 8-2=6 \$\$ 4*6=24
2-3=-1 \$\$ 5--1=6 \$\$ 4*6=24
3+5=8 \$\$ 4+8=12 \$\$ 2*12=24
4+3=7 \$\$ 7+5=12 \$\$ 2*12=24
4+5=9 \$\$ 9+3=12 \$\$ 2*12=24
5+4=9 \$\$ 3+9=12 \$\$ 2*12=24
3-2=1 \$\$ 1+5=6 \$\$ 4*6=24
2-5=-3 \$\$ 3--3=6 \$\$ 4*6=24
4+3=7 \$\$ 5+7=12 \$\$ 2*12=24
5+3=8 \$\$ 8+4=12 \$\$ 2*12=24
5+3=8 \$\$ 4+8=12 \$\$ 2*12=24
5-2=3 \$\$ 3+3=6 \$\$ 4*6=24
5+4=9 \$\$ 9+3=12 \$\$ 2*12=24
3+4=7 \$\$ 7+5=12 \$\$ 2*12=24
4+5=9 \$\$ 3+9=12 \$\$ 2*12=24

请依次输入四个数(输入quit退出): 6 4 5 8

4-6=-2 \$\$ 5+-2=3 \$\$ 8*3=24
4-6=-2 \$\$ -2+5=3 \$\$ 8*3=24
6-4=2 \$\$ 5-2=3 \$\$ 8*3=24
6-5=1 \$\$ 4-1=3 \$\$ 8*3=24
4+5=9 \$\$ 9-6=3 \$\$ 8*3=24
5-6=-1 \$\$ -1+4=3 \$\$ 8*3=24
5+4=9 \$\$ 9-6=3 \$\$ 8*3=24
5-6=-1 \$\$ 4+-1=3 \$\$ 8*3=24

请依次输入四个数(输入quit退出): quit

第三题代码二

In [9]:

```
import itertools

def get_wei24(cards):
    for nums in itertools.permutations(cards): # 获取四个数所有排列
        for ops in itertools.product('+-*/', repeat=3): # 获取三个运算符所有排列
                                                    # (每个运算符可重复三次)

            # 构造三种中缀表达式 (bsd)
            bds1 = '({0} {4} {1}) {5} ({2} {6} {3})'.format(*nums, *ops) # (a+b)*(c-d)
            bds2 = '(({0} {4} {1}) {5} {2}) {6} {3}'.format(*nums, *ops) # (a+b)*c-d
            bds3 = '{0} {4} ({1} {5} ({2} {6} {3}))'.format(*nums, *ops) # a/(b-(c/d))

            for bds in [bds1, bds2, bds3]: # 遍历
                try:
                    if abs(eval(bds) - 24.0) < 1e-10: # eval函数
                        return bds
                except ZeroDivisionError: # 零除错误!
                    continue

    return 'none'
```

In [10]:

```
while True:
    string = input('请依次输入四个数(输入quit退出): ')
    if string=='quit':
        break
    try:
        pz = [int(i) for i in string.split()]
    except:
        continue
    print(get_wei24(pz))
```

请依次输入四个数(输入quit退出): 1 2 3 4

((1+2)+3)*4

请依次输入四个数(输入quit退出): 4 5 6 7

4*(5-(6-7))

请依次输入四个数(输入quit退出): 2 5 7 1

2*(5+(7*1))

请依次输入四个数(输入quit退出): 23 45 12 34

(23+45)*(12/34)

请依次输入四个数(输入quit退出): quit