华东师范大学计算机科学技术系作业

	华东师范大学计算机科学技术系作业	
课程名称：编程导论 Python	年级：2018级	作业成绩：
指导教师：杨燕	姓名：吴子靖	提交作业日期：2018年11月7日
专业：计算机系	学号：10185102141	作业编号： 5

一、 请利用所学知识，将下面的“不完美函数”改写成完美函数。

```
#<程序：“不完美函数”>
res
def add(a,b):
    a = a*b; res = a+b
add(2,3)
print("最终结果为：",res)
```

（10分）

In [16]:

```
def add(a,b):
    a=a*b
    res=a+b
    return res
result=add(2,3)
print("最终结果为:",result)
```

最终结果为: 9

二、 改写本章 3.1.2 部分<程序：参数与返回值举例>中的 find 函数，使其可以实现新的功能：查找序列中是否有字符‘f’，若有，则返回 True 与一个 列表，列表中记录所有字符 f 所在的索引；若无，则返回 False 与空列表。
例如，对于'abeffestffe'；返回 True, [3,4,8,9]。
例如，对于[23,4,6,'e']；返回 False, []。
（10分）

In [11]:

```
def find(str1,str2):
    i=0;L=[]
    while i<len(str1):
        if str1[i]==str2:
            L.append(i)
        i=i+1
    if L==[]:
        return (False,L)
    else:
        return (True,L)
a,b=find("abeffestffe","f")
c,d=find([23,4,6,"e"],"f")
print(a,b)
print(c,d)
```

```
True [3, 4, 8, 9]
False []
```

三、这个程序，将会输出什么？在 g-func()中哪些是局部变量?

```
#<程序：局部变量与全局变量举例>
b, c=2, 4
def g_func(d):
    global a ; a=d*c
g_func(b) ; print(a)
```

（10分）

In [18]:

```
b,c=2,4
def g_func(d):
    global a
    a=d*c
g_func(b)
print(a)
#a使用了global函数，b，c在函数外的等号左侧，所以abc均为全局变量，而d是函数内的变量，所以只有d是局部变量
```

```
8
```

四、局部与全局变量练习。请分析<程序：四则运算例子>的执行过程，并说明输出结果。

```
#<程序：四则运算例子>
def do_div(a, b):
    c=a/b                #a, b, c 都是 do_div()中的局部变量
    print (c); return c
def do_mul(a, b):
    global c ; c=a*b     #a, b 是 do_mul()的局部变量，c 是全局变量
    print (c) ; return c
def do_sub(a, b):
    c=a-b                 #a, b, c 都是 do_sub()中的局部变量
    c=do_mul(c, c)
    c=do_div(c, 2)
    print (c); return c
def do_add(a, b):        #参数 a 和 b 是 do_add()中的局部变量
    global c
    c=a + b              #全局变量 c，修改了 c 的值
    c=do_sub(c, 1)       #再次修改了全局变量 c 的值
    print (c)
#所有函数外先执行:
a=3                      #全局变量 a
b=2                      #全局变量 b
c=1                      #全局变量 c
do_add(a, b)             #全局变量 a 和 b 作为参数传递给 do_add()
print (c)                #全局变量 c
```

（10分）

执行do_add函数，c=1是全局变量，c=a+b=5 再执行do_sub函数，此时的c是局部变量，为方便说明，记录为c1 c1=c-1=4,此时再执行do_mul函数，c=c1*c1=44*=16,输出此时的c=16，并将c=16返回给c1 然后执行do_div函数，局部变量c2=c1/2=8,输出此时的c2=8,并将c2=8返回给c1 然后再回到do_sub函数，再次输出c1=8，并将c1=8返回给全局变量c 然后回到do_add函数，再次输出c=8 然后执行最后的print语句，输出c=8

In [19]:

```
def do_div(a,b):
    c=a/b
    print(c);return c
def do_mul(a,b):
    global c;c=a*b
    print(c);return c
def do_sub(a,b):
    c=a-b
    c=do_mul(c,c)
    c=do_div(c,2)
    print(c);return c
def do_add(a,b):
    global c
    c=a+b
    c=do_sub(c,1)
    print(c)
a=3;b=2;c=1
do_add(a,b)
print(c)
```

```
16
8.0
8.0
8.0
8.0
```

五、修改习题 3.4 中的<程序：四则运算例子>，去掉 do_add()中的 global c 语句，分析程序将会输出什么?

（10分）

执行do_add函数，c=1是全局变量，此时的局部变量c1=a+b=5 再执行do_sub函数，局部变量c2=c1-1=4 此时再执行do_mul函数，全局变量c=c2*c2=4*4=16,输出此时的c=16，并将c=16返回给c2 然后执行do_div函数，局部变量c3=c2/2=8,输出此时的c3=8，并将c3=8返回给c2 然后再回到do_sub函数，再次输出c2=8，并将c2=8返回给c1 然后回到do_add函数，再次输出c1=8 然后执行最后的print语句，输出全局变量c=16

In [20]:

```
def do_div(a,b):
    c=a/b
    print(c);return c
def do_mul(a,b):
    global c;c=a*b
    print(c);return c
def do_sub(a,b):
    c=a-b
    c=do_mul(c,c)
    c=do_div(c,2)
    print(c);return c
def do_add(a,b):
    c=a+b
    c=do_sub(c,1)
    print(c)
a=3;b=2;c=1
do_add(a,b)
print(c)
```

```
16
8.0
8.0
8.0
16
```

六、嵌套函数中局部与全局变量的练习。分析<程序：嵌套函数局部与全 局变量练习>，每个变量分别是局部变量还是全局变量，同时说出打印结果。

```
#<程序：嵌套函数局部与全局变量练习>
a=1;b=2
def fun(x):
    def F():
        global a ;  a=x+y+b
        return a
    y=12 ; x=x+2 ; a=F()
fun(b)
print("Finally, a is: %d and b is: %d"%(a,b))
```

（10分）

a,b在函数之外，是全局变量，x，y是函数内的等号左侧变量，是局部变量 先执行fun（b）函数，y=12,局部变量b=全局变量b+2=2+2=4，函数内的b的值是4，但是函数外的b的值并没有被改变 a=局部变量b+y+全局变量b=2+12+4=18，因为global的原因，函数内外的a的值均变为了18 所以最终的输出结果，a=18，b仍=2

In [22]:

```
a=1;b=2
def fun(x):
    def F():
        global a;a=x+y+b
        return a
    y=12;x=x+2;a=F()
fun(b)
print("Finally, a is: %d and b is: %d"%(a,b))
```

```
Finally, a is: 18 and b is: 2
```

七、假设一个列表为 L，则 L.reverse() 和 L[-1:-1-len(L):-1] 的差别在哪里？（10分）

L.reverse()是将列表L本身进行反转，是没有返回值的 而L[:-1:-1-len(L)-1:-1]是将L中的元素逆向遍历一遍，并依次复制到一个新列表中，是具有返回值的，列表本身没有改变

In [8]:

```
#L.reverse()是将列表L本身进行反转，是没有返回值的
L=[1,2,3,4]
a=L.reverse()
print(a)
print(L)
```

```
None
[4, 3, 2, 1]
```

In [9]:

```
#而L[:-1:-1-len(L)-1:-1]是将L中的元素逆向遍历一遍，并依次复制到一个新列表中，是具有返回值的，列表本身没有改变
L=[1,2,3,4]
a=L[-1:-1-len(L):-1]
print(a)
print(L)
```

```
[4, 3, 2, 1]
[1, 2, 3, 4]
```

八、假设一个列表为 L ，我们知道 L.remove(x) 是除去 L 中第一个值为 x 的元素，那么要除去 L 中所有是 x 的元素，要怎么办？（10分）

In [1]:

```
#先遍历一遍L，查出x的数量，再用n次remove
L=[1,2,3,4,1,5,6,7,1,8,43,1,97,1,571,1]#假设去掉所有1的元素
i=0
for e in L:
    if e==1:
        i=i+1
while i!=0:
    L.remove(1)
    i=i-1
print(L)
```

```
[2, 3, 4, 5, 6, 7, 8, 43, 97, 571]
```

九、如何用 L.insert(i,x) 实现 L.append(x)？（10分）

In [2]:

```
#令i=len(L),例如
L=[1,2,3,4,5,6,7,8]
L1=L[:]
L.append(9)
L1.insert(len(L),9)
print(L)
print(L1)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9]
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

十、利用 for 循环将一个字符串列表的双重倒转。给定一个字符串列表， 将整个序列倒转，同时每个字符串元素也要倒转，输出倒转后的列表。
比如 L=['It is','very very','funny','!']； 则完全倒转的结果为 L_new=['!','ynnuf','yrev yrev','si tI']。（10分）

In [1]:

```
L=["It is","very very","funny","!"]
i=0
while i<len(L):
    L[i]=L[i][::-1]
    i=i+1
L_new=L[::-1]
L_new
```

Out[1]:

```
['!', 'ynnuf', 'yrev yrev', 'si tI']
```