

## 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10月24日
专业：计算机系	学号：10185102141	作业编号：3

一、请用自定义函数编程实现  $f(n)$ , 函数功能为：求  $n$  的阶乘。利用函数  $f(n)$ , 编程实现对任意给定的3个整数  $x \leq y \leq z$ , ( $1 < x \leq y \leq z < 100$ ), 求  $x! + y! + z!$ 。(20分)

In [12]:

```
def f(n):  
    i=1  
    a=1  
    while i<=n:  
        a=a*i  
        i=i+1  
    return a  
x=int(input("x的值="))  
y=int(input("y的值="))  
z=int(input("z的值="))  
print(f(x)+f(y)+f(z))
```

x的值=5  
y的值=7  
z的值=6  
5880

二、请用自定义函数编程实现  $\text{factor}(\text{num}, k)$ , 函数功能为：求整数  $\text{num}$  中包含因子  $k$  的个数，如果没有该因子则返回0。例如  $12=2*2*3$ , 则  $\text{factors}(12,2)=2$ ,  $\text{factors}(12,3)=1$ ,  $\text{factors}(12,4)=1$ ,  $\text{factors}(12,5)=0$ 。(20分)

In [13]:

```
def factor(num, k):
    i=0
    while (num%k)==0:
        num=num/k
        i=i+1
    return i
num=int(input("整数num="))
k=int(input("因子k="))
print("包含因子个数为", factor(num, k))
```

整数num=12

因子k=2

包含因子个数为 2

三、请用自定义函数编程实现pai(e)，函数功能为：根据以下公式求 $\pi$ 的值，直到某一项的值小于给定的精度e。

$$\frac{\pi}{2} = 1 + \frac{1}{3} + \frac{1}{3} * \frac{2}{5} + \frac{1}{3} * \frac{2}{5} * \frac{3}{7} + \frac{1}{3} * \frac{2}{5} * \frac{3}{7} * \frac{4}{9} + \dots$$

输入：精度e，例如0.0005

输出： $\pi$ 的值，上例为3.14058 （20分）

In [24]:

```
def pai(e):
    a=1
    i=3
    j=1
    b=j/i
    while b>=e:
        a=a+b
        i=i+2
        j=j+1
        b=b*j/i
    pi=2*a
    return pi
e=float(input("精度e="))
pai(e)
```

精度e=0.0005

Out[24]:

3.140578169680336

四、请用自定义函数编程实现fun(a,b)，函数功能为：将两个三位数的整数a，b合并成一个整数c，合并规则为c的十万位、千位和十位分别是a的百位、十位和个位，而c的万位、百位和个位分别是b的百位、十位和个位。例如：输入为 a=123，b=456，则输出为 c=142536 （20分）

In [25]:

```
def fun(a, b):
    A=str(a)
    B=str(b)
    c=(int(A[0]))*(10**5)+(int(B[0]))*(10**4)+(int(A[1]))*(10**3)+(int(B[1]))*(10**2)+(
int(A[2]))*(10**1)+(int(B[2]))*(10**0)
    return int(c)
a=int(input("a="))
b=int(input("b="))
fun(a, b)
```

a=123

b=456

Out[25]:

142536

五、请以多项式 $4x^{20} + 3x^{10} + 5$ 和多项式 $2x^8 + x - 1$ 为例，使用本章中子列表形式的数据结构（第三种），编写多项式乘法和除法的Python程序。（20分）

In [20]:

```
def add_poly(L1,L2):
    R=L1[:]
    for i in range (len(L2)):
        for j in range (len(R)):
            if R[j][0]==L2[i][0]:
                R[j][1]=R[j][1]+L2[i][1]
                if R[j][1]==0:
                    R.remove(R[j])
                break
            else:
                R.append(L2[i])
    return R
def subtract_poly(L1,L2):
    L2=L2[:]
    for i in range (len(L2)):
        L2[i][1]=-(L2[i][1])
    return (add_poly(L1,L2))
def multiply_poly(L1,L2):
    if len(L1)>len(L2):
        L1,L2=L2,L1
    R=[];zeros=[]
    for i in range (len(L1)):
        T=zeros[:]
        for e in L2:
            T.append([e[0]+L1[i][0],e[1]*L1[i][1]])
        R=add_poly(R,T)
    return R
def max1(L1):
    for i in range (0,len(L1)-1):
        if L1[i][0]>L1[i+1][0]:
            Max=i
```

```

        break
    else:
        Max=i+1
    return Max
def divide_poly1(L1,L2):
    S=[]
    T=L1[:]
    while T[max1(T)][0]>=L2[0][0]:
        a=T[max1(T)][0]-L2[0][0]
        b=T[max1(T)][1]/L2[0][1]
        c=[a,b]
        S.append(c)
        d=multiply_poly([c],L2)
        T=subtract_poly(T,d)
    return S,T
def order(L1):
    R=[]
    T=L1[:]
    while len(T)>1:
        R.append(T[max1(T)])
        T.remove(T[max1(T)])
    R.append(T[0])
    return R
L1=[[20,4],[10,3],[0,5]]
L2=[[8,2],[1,1],[0,-1]]
A=multiply_poly(L1,L2)
b=divide_poly1(L1,L2)
B=order(b[0])
C=order(b[1])
print("积:",A)
print("商:",B)
print("余数:",C)

```

积: [[28, 8], [21, 4], [20, -4], [18, 6], [11, 3], [10, -3], [8, 10], [1, 5], [0, -5]]

商: [[12, 2.0], [5, -1.0], [4, 1.0], [2, 1.5]]

余数: [[6, 1.0], [5, -2.0], [4, 1.0], [2, 1.5], [0, 5], 3, -1.5]