**R6-1 使用函数求素数和 (10 分)**

使用函数求素数和

prime(p), 其中函数prime当用户传入参数p为素数时返回True，否则返回False. PrimeSum(m,n),函数PrimeSum返回区间[m, n]内所有素数的和。题目保证用户传入的参数1<=m<n。

函数接口定义：

在这里描述函数接口：

prime(p)，返回True表示p是素数，返回False表示p不是素数

PrimeSum(m,n)，函数返回素数和

裁判测试程序样例：

/\* 请在这里填写答案 \*/

m,n=input().split()

m=int(m)

n=int(n)

print(PrimeSum(m,n))

输入样例：

在这里给出一组输入。例如：

1 10

输出样例：

在这里给出相应的输出。例如：

17

def PrimeSum(m,n):

sum = 0

if m==1:

m += 1

for i in range(m,n):

for j in range(2,i):

if i % j==0:

break

else:

sum+=i

return sum

**R6-2 缩写词 (10 分)**

缩写词是由一个短语中每个单词的第一个字母组成，均为大写。例如，CPU是短语“central processing unit”的缩写。

函数接口定义：

acronym(phrase);

phrase是短语参数，返回短语的缩写词

裁判测试程序样例：

/\* 请在这里填写答案 \*/

phrase=input()  
print(acronym(phrase))

输入样例：

central processing unit

输出样例：

CPU

def acronym(phrase):

p= phrase.split()

ret = ""

for i in p:

ret += i[0].upper()

return ret

**R6-3 jmu-python-计算薪资 (10 分)**

某公司销售员底薪为5000，销售业绩与利润提成的关系如下表所示（计量单位：元）



编写函数，计算员工月薪。

函数接口定义：

bonus(sales)

其中 sales 是参数，表示员工的月销售业绩。

裁判测试程序样例：

/\* 请在这里填写答案 \*/

sales=eval(input())

print("%.2f"%bonus(sales))

输入样例1：

50800

输出样例1：

15160.00

输入样例2：

35000

输出样例2：

10250.00

def bonus(sales):

s = float(sales)

n =5000

if s <= 10000:

return n

elif 10000<s<=20000:

return s\*0.1 + n

elif 20000<s<=50000:

return s\*0.15 + n

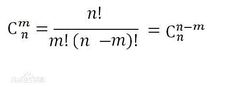
elif 50000<s<=100000:

return s\*0.2 + n

else:

return s\*0.25 + n

**R6-4 python函数-组合数 (10 分)**

请编写函数完成：从n个元素中取出m个元素的组合数：参考公式  。输入要求：n和m均为正整数；如果输入非法数据请参照下方输入输出样例的警告信息；

函数接口定义：

定义两个函数，函数名及参数要求如下：

Fun\_Fact(x) #定义阶乘函数，对x求阶乘

Fun\_Comb ( m, n) #调用阶乘函数，使用题目公式，再设计函数计算组合数

其中'x' 、'n' 和 'm'都是用户传入的参数。x、 n和m 的值均为正整数；且 n>=m；第二个函数Fun\_Comb ( n, m ) 须返回组合数个数。

裁判测试程序样例：

*/\* 请在这里填写答案 \*/*

a,b = input().split(',')

**if** a.isdecimal() **and** b.isdecimal() **and** (**int**(a)<=**int**(b)):

Comb\_Result = Fun\_Comb(**int**(a), **int**(b))

print("result={:.2f}".format(Comb\_Result))

**else**:

print("Error data,please input again!")

输入样例1：

2,7

输出样例1：

result=21.00

输入样例2：

4,-10

输出样例2：

Error data,please input again!

def Fun\_Fact(x):

s = 1

while(x>1):

s \*= x

x -= 1

return s

def Fun\_Comb(m,n):

return Fun\_Fact(n)/(Fun\_Fact(m)\*Fun\_Fact(n-m))

**R6-1 jmu-python-函数-找钱 (20 分)**

买单时，营业员要给用户找钱。营业员手里有10元、5元、1元(假设1元为最小单位)几种面额的钞票，其希望以  
尽可能少(张数)的钞票将钱换给用户。比如，需要找给用户17元，那么其需要给用户1张10元，1张5元，2张1元。  
而不是给用户17张1元或者3张5元与2张1元。

### 函数接口定义：

giveChange(money) *#money为要找的钱。经过计算，应按格式"要找的钱 = x\*10 + y\*5 + z\*1"输出。*

### 裁判测试程序样例：

/\* 请在这里填写答案 \*/

n = int(input())

**for** i **in** range(n):

giveChange(int(input()))

### 输入样例：

5

109

17

10

3

0

### 输出样例：

109 = 10\*10 + 1\*5 + 4\*1

17 = 1\*10 + 1\*5 + 2\*1

10 = 1\*10 + 0\*5 + 0\*1

3 = 0\*10 + 0\*5 + 3\*1

0 = 0\*10 + 0\*5 + 0\*1

def giveChange(money):

x=int(money/10)

m=money%10

y=int(m/5)

m=m%5

z=int(m/1)

print("{} = {}\*10 + {}\*5 + {}\*1".format(money,x,y,z))

**R6-2 使用函数输出指定范围内Fibonacci数的个数 (20 分)**

本题要求实现一个计算Fibonacci数的简单函数，并利用其实现另一个函数,输出两正整数m和n（0<m<n≤100000）之间的所有Fibonacci数的数目。 所谓Fibonacci数列就是满足任一项数字是前两项的和（最开始两项均定义为1）的数列,fib(0)=fib(1)=1。其中函数fib(n)须返回第n项Fibonacci数；函数PrintFN(m,n)用列表返回[m, n]中的所有Fibonacci数。

### 函数接口定义：

在这里描述函数接口。例如：

fib(n),返回fib(n)的值

PrintFN(m,n)，用列表返回[m, n]中的所有Fibonacci数。

### 裁判测试程序样例：

在这里给出函数被调用进行测试的例子。例如：

/\* 请在这里填写答案 \*/

m,n,i=input().split()

n=int(n)

m=int(m)

i=int(i)

b=fib(i)

print("fib({0}) = {1}".format(i,b))

fiblist=PrintFN(m,n)

print(len(fiblist))

### 输入样例：

在这里给出一组输入。例如：

20 100 6

### 输出样例：

在这里给出相应的输出。例如：

fib(6) = 13

4

def fib(n):

a,b,i=0,1,0

while i<n:

b=b+a

a=b-a

i+=1

return b

def PrintFN(m,n):

list=[]

for i in range(n):

x=fib(i)

if m<=x<=n:

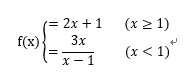
list.append(x)

elif x>n:

break

return list

**R6-3 函数-数学公式 (10 分)**

请根据数学公式  编写函数，用户输入数据后（输入格式为浮点数），调用该函数计算后返回对应的值。

### 函数接口定义：

函数接口定义如下：

Funx(x)

其中 x 是用户传入的参数。函数须返回对应的值，格式为浮点数，精度保留小数点后两位。

### 裁判测试程序样例：

*/\* 请在这里填写答案 \*/*

thisx = eval(input())

print("f({}) = {:.2f}".format(thisx,Funx(thisx)))

### 输入样例：

在这里给出一组输入。例如：

0.5

### 输出样例：

在这里给出相应的输出。例如：

f(0.5) = -3.00

def Funx(x):

if x>=1:

y=2\*x+1

else:

y=3\*x/(x-1)

return y

**R7-1 计算N阶乘 (10 分)**

自己创建一个函数，传入一个参数n，返回n的阶乘结果。

### 输入格式:

输入大于0的任意整数，如：6。

### 输出格式:

显示阶乘的结果。

### 输入样例:

：

6

### 输出样例:

720

def jiecheng(n):

if n == 1:

return 1

elif n == 0:

return 0

num = 1

for i in list(range(1,n+1)):

num \*= i

return num

n=eval(input())

print(jiecheng(n))

**R7-2 编写函数统计大写和小写字母数量 (10 分)**

编写一个函数，能对传递给函数的任意字符串中的大写和小写字母进行统计，并将结果存于列表中，并打印出结果。

### 输入格式:

通过键盘输入任意字符串。

### 输出格式:

输出列表格式的结果，只有两个数据项，第一项为大写字母的统计结果，第二项为小写字母的统计结果。

### 输入样例:

aaaabbbbC

### 输出样例:

[1, 8]

def func(n):

upper=0

lower=0

for i in a:

if i.isupper():

upper+=1

elif i.islower():

lower+=1

return upper,lower

a=input()

b=func(a)

print(list(b))

**R6-1 判断回文 (10 分)**

定义判断字符串是否为回文的函数，然后调用该函数判断从键盘输入的任意字符串是否为回文，若是则打印"Yes!"，否则打印"No!"。

### 函数接口定义：

在这里描述函数接口。例如：

huiwen(n);

其中 n 是用户传入的参数。

### 裁判测试程序样例：

在这里给出函数被调用进行测试的例子。例如：

/\* 请在这里填写答案 \*/

x = input()

print(huiwen(x))

### 输入样例1：

在这里给出一组输入。例如：

abcdcba

### 输出样例1：

在这里给出相应的输出。例如：

Yes

### 输入样例2：

在这里给出一组输入。例如：

abcdcb1

### 输出样例2：

在这里给出相应的输出。例如：

No

def huiwen(n):

if n==n[::-1]:

return "Yes"

else:

return "No"

**R6-2 编写函数统计两个字符串相同的字符数 (10 分)**

设计一个函数，使其能统计两个字符串相同的字母数量，功能要求：输入字符串1和字符串2，如果两个字符串长度相同，经过函数处理要求统计第2个字符串与第1个字符串相同位置具有相同字符的数量。如果输入的字符串2与字符串1长度不匹配，则报错"Sorry. the string length is not same!"，并输出：None。

### 函数接口定义：

在这里描述函数接口。例如：

compare ( str1, str2 );

其中 str1 和 str2 都是用户传入的参数。

### 裁判测试程序样例：

在这里给出函数被调用进行测试的例子。例如：

/\* 请在这里填写答案 \*/

x = input()

y = input()

print(compare(x,y))

### 输入样例1:

例如：

Chengdu University of information Techenology

Chengdu University Of Information Techenology

### 输出样例1:

例如：

43

### 输入样例2:

例如：

Chengdu University

Chengdu

### 输出样例2:

例如：

Sorry. the string length is not same!

None

def compare(str1,str2):

m = len(str1)

n = len(str2)

s = 0

if m == n:

for i in range(len(str1)):

if str1[i] == str2[i]:

s += 1

return s

else:

print("Sorry. the string length is not same!")

return

**R6-3 函数-求n!递归函数计算阶乘和 (10 分)**

请根据用户输入的数字n，编写函数实现阶乘累加值：1!+3!+5!+7!+9!+...+(2n-1)!，具体要求：1.分别编写两个函数实现阶乘和奇数数列的阶乘累加和并按照格式输出；2.阶乘函数采用递归方式计算，阶乘累加函数的输出内容和格式要与输出样例一致。

### 函数接口定义：

函数接口如下：

Fun\_fact(n) #递归阶乘函数接口

Fact\_output(m) #阶乘的累加函数接口

接口参数：其中 m 是用户传入的参数，表示需要累加的范围,且为自然数。 n 为需要计算的阶乘参数； 函数1需要返回 n 的阶乘结果，函数2需要返回阶乘累加字符串样式和累加值(具体内容请参考输出样例1)。

### 裁判测试程序样例：

/\* 请在这里填写答案 \*/

factx = **int( input(**))

resultx,resulty = Fact\_ou**tput(**factx)

s\_format = "{} = {}".**format**(resultx,resulty)

p**rint(**s\_format)

### 输入样例1：

输入

5

### 输出样例1：

输出：

1!+3!+5!+7!+9! = 368047

def Fun\_fact(n):

if n == 1:

return 1

else:

s = 1

for i in range(2,n+1):

s = s\*i

return s

def Fact\_output(m):

summ = 0

l = []

for i in range(1,2\*m):

if i%2 == 0:

summ = summ+0

else:

l.append(str(i)+"!")

summ = summ + Fun\_fact(i)

return "+".join(l) ,sum