# 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：2019年1月2日 |
| 专业：计算机系 | 学号：10185102141 | 作业编号： 12 |

## 一、编程实现随机产生1个20位的数，使得该数与111这个数互质。（20分）

In [1]:

```
f = True
a = int(111**(1/2))+1  #判断111是否是质数
for i in range(2,a):
    if 111%a == 0:
        f = False
        break
print(f)
```

```
True
```

In [1]:

```
#通过上述的判断，知道111是一个质数，因此只要这个20位的数对111取余不为零，这个数就和111互质
x1 = pow(10,19)//111
x2 = pow(10,20)//111
import random
m = random.randint(x1+1,x2)
n = random.randint(1,110)
result = m*111 + n
print("随机产生的数字是：%d"%result)
print("%d  的长度是：%d"%(result,len(str(result))))
print("%d  对111取余的结果是：%d"%(result,result%111))
```

```
随机产生的数字是：99743458894260620963
99743458894260620963  的长度是：20
99743458894260620963  对111取余的结果是：41
```

## 二、写Python程序算出$a^x mod b$的值。函数mod(a, x, b)返回$a^x mod b$的值。假设$a$和$b$都是最多为10位的整数，而$x$可以是最多为200位的整数。请用递归编写此程序。（20分）

In [1]:

```
def mod(a,x,b):
    if x == 1:
        return a % b
    if x == 0:
        return 1 % b
    return (mod(a,x//2,b)*mod(a,x-x//2,b))%b
print("mod(3,32,7) = %d"%mod(3,32,7))
print("mod(4,32,7) = %d"%mod(4,32,7)) #对书上例题的验证
```

```
mod(3,32,7) = 2
mod(4,32,7) = 2
```

三、世界上常用的一种安全编码方式为RSA，其中产生公钥和私钥的过程中会用到本章介绍的倒数的概念。其实现方式为：给定两个质数$p$，$q$，随机产生一个奇数$e$，满足$e < (p-1)(q-1)$，而且$e$与$(p-1)(q-1)$互质，即$gcd(e,(p-1)(q-1)) = 1$。在$e$的基础上产生$e$的倒数$d$，即$e \cdot d = 1(mod(p-1)(q-1))$。以上过程中产生的$e$即为公钥，$d$即为私钥。
请编程实现求解私钥：对于给定的两个质数$p = 128543041447753$和$q = 1062573853363145487845851$，先随机产生$e < (p-1)(q-1$并且满足$gcd(e,(p-1)(q-1)) = 1$，然后求$d$并打印出来。（20分）

In [6]:

```
def gcd(x,n): #最大公因数
    if x < n:
        x,n = n,x
    while x % n != 0:
        t = n
        n = x % n
        x = t
    return n
def ex(x,y,vx,vy): #求解逆元
    r = x % y;z = x//y
    if r == 0:
        return(y,vy)
    vx[0] = vx[0] - z*vy[0]
    vx[1] = vx[1] - z*vy[1]
    return ex(y,r,vy,vx)
def mod(e,n):
    vx = [1,0];vy = [0,1]
    if e > n:
        G,X = ex(e,n,vx,vy)
        d = X[0] % n
    else:
        G,X = ex(n,e,vx,vy)
        d = X[1] % n
    return d


p=128543041447753;q=106257385363145487845851
res = (p-1)*(q-1)
import random
e = random.randint(1,res-1)
while gcd(e,res) != 1:  #随机产生一个与(p-1)(q-1)互质的数
    e = random.randint(1,res-1)
print("产生的随机数是: %d"%e)
print("%d与(p-1)(q-1)的最大公因数是%d"%(e,gcd(e,res)))
d = mod(e,res)
print("私匙d = %d"%d)
```

```
产生的随机数是: 560910114906020669922637102675244593 79
560910114906020669922637102675244593 79与(p-1)(q-1)的最大公因数是1
私匙d = 142184230298638339375154886434354530 19
```

四、请将<程序：combination_3>的代码改成可以处理L有重复元素的情况。（20分）

In [9]:

```
import numpy as np
def main():
    def combination_3(L,k):
        if len(L) <= k:
            return [L]
        if k == 0:
            return [[]]
        T1 = combination_3(L[0:len(L)-1],k-1)
        T2 = combination_3(L[0:len(L)-1],k)
        T  = []
        for e in T1:
            e.append(L[len(L)-1])
            T.append(e)
        return (T2+T)
    result = combination_3(L,k)
    arr = np.array(result)
    print(np.array(list(set([tuple(t) for t in arr]))))
L = [1,2,3,3,5]
k = 3
main()
```

```
[[2 3 5]
 [2 3 3]
 [3 3 5]
 [1 2 3]
 [1 3 5]
 [1 2 5]
 [1 3 3]]
```

五、改写<程序：combination_4>。使得从L的第一个元素开始考虑，而不是从最后一个开始考虑。也就是说首先考虑一定选择第一个数L[0]，然后从剩下的$n-1$个数中选$k-1$个数的组合了然后，一定不选L[0]，一定选第二个数L[1]，再从剩下的$n-2$个数中选取$k-1$个数的组合；以此类推。（20分）

In [97]:

```
def combination_4(L,k):
    if k == 0 or len(L) < k:
        return [[]]
    if len(L) == k:
        return [L]
    n = len(L);T = [L[1]];R = []
    for i in range(k+1):
        A = combination_4(L[i:],k-1)
        for e in A:
            e_new = e + T
            e_new.sort()
            if not e_new in R:
                R.append(e_new)
        T = [L[0]]
    return R
L = [1,2,3,4,5]
k = 3
print(combination_4(L,k))
```

```
[[2, 2, 2], [1, 2, 2], [1, 2, 3], [1, 2, 4], [1, 3, 3], [1, 2, 5], [1, 4, 4], [1,
3, 4], [1, 3, 5], [1, 4, 5]]
```