

苏州大学实验报告

| | | | | | | | |
|------|-------------|-------|-------|------|----|----|------------|
| 院、系 | 计算机学院 | 年级专业 | 计算机科学 | 姓名 | 柯骅 | 学号 | 2027405033 |
| 课程名称 | Python 程序设计 | | | | | 成绩 | |
| 指导教师 | 李正华 | 同组实验者 | 无 | 实验日期 | | | |

实 验 名 称 实验五 循环结构程序设计

一. 实验目的

通过本次实验要达到如下目的:

1. 掌握循环结构程序设计方法
2. 掌握 continue 和 break 语句
3. 掌握 for 语句和 while 语句以及嵌套结构的使用方法
4. 逐步建立起组织复杂程序逻辑和流程的能力

二. 实验内容

1. 【for 语句】用 * 输出一个等腰三角形。提示用户输入一个整数 n, 代表输出的等边三角形由 n 行 * 组成。例: 输入 n=3。输出:

```
  *
 ***
*****
```

2. 【for 语句】输出一个乘法表。要求输入一个整数 n, 输出 n*n 的乘法表, 乘法表打印出来为下三角样式, 格式工整。例: 输入 n=4。输出:

```
      1    2    3    4
1    1
2    2    4
3    3    6    9
4    4    8   12   16
```

提示: 可以使用 print(i, end='t') 或 print('%10d'% i) 控制输出的数据格式。

3. 【for 语句/while 语句】用 * 输出一个正六边形, 输入一个整数 n 代表输出的正六边形的边的长度(*的数目)。例: 输入 n=3。输出:

```
  *   *   *
 *   *   *   *
*   *   *   *   *
 *   *   *   *
  *   *   *
```

4. 【for 语句/while 语句】循环提示用户输入一个整型数字 n(n 代表后续需要输入整型数的数量), 将 n 个整型数加起来并输出, 如果输入的是非整型数则提示当前的输入非法需要重新输入数值, 如果输入 'n=0' 代表退出程序, 否则继续提示用户输入新的 n。

例:

Please input the number of numbers: (假设输入 n=3)

Please input number 1: (假设输入 3)

Please input number 2: (假设输入 4)

Please input number 3: (假设输入 5)

输出: sum = 12

Please input the number of numbers:

...

Please input the number of numbers: (假设输入 n=0, 则退出程序)

5. 【for 语句/while 语句】提示用户输入一个整数 n, 然后输出 [1, n) 内的所有的素数。提示: 质数 (prime number) 又称素数, 有无限个。质数定义为在大于 1 的自然数中, 除了 1 和它本身以外不再有其他因数的数称为质数。例: 输入 n = 10。输出: 2, 3, 5, 7
6. 【for 语句/while 语句】求 $S_n = a + aa + \dots + \overbrace{aa \dots a}^n$ 的值。其中 a 是一个数字。a 和 n 都是由键盘输入。例如: 求 $S = 2 + 22 + 222 + 2222 + 22222 + 222222$, 那么 a=2 且 n=6。
7. 矩阵相加: 提示用户输入一个数字 n, 为矩阵的行数, 再提示用户输入一个数字 m, 为矩阵的列数, 接下来, 提示用户输入 2*n*m 个数字 (每次输入一个数字)。输出 C=A+B。
提示: 思考怎么用 Python 实现二维数组 (如果做矩阵相加没有问题了, 可以思考如何做矩阵相乘)。

例: 输入:

Please input the number of rows: (假设输入 n=2)

Please input the number of columns: (假设输入 m=3)

Please input A[0,0]: 1

Please input A[0,1]: 1

Please input A[0,2]: 1

Please input A[1,0]: 1

Please input A[1,1]: 1

Please input A[1,2]: 1

Please input B[0,0]: 2

Please input B[0,1]: 2

Please input B[0,2]: 2

Please input B[1,0]: 2

Please input B[1,1]: 2

Please input B[1,2]: 2

输出: C = [[3, 3, 3], [3, 3, 3]]

8. 矩有 n 个小朋友围成一圈玩游戏, 小朋友从 1 至 n 编号, 2 号小朋友坐在 1 号小朋友的顺时针方向, 3 号小朋友坐在 2 号小朋友的顺时针方向,, 1 号小朋友坐在 n 号小朋友的顺时针方向。

从 1 号小朋友开始顺时针报数, 接下来每个小朋友的报数是上一个小朋友报的数加 1。若一个小朋友报的数为 k 的倍数, 则该小朋友被淘汰出局, 不再参加以后的报数。当游戏中只剩下一个小朋友时, 该小朋友获胜。

例如, 当 n=5, k=2 时:

1 号小朋友报数 1;

2 号小朋友报数 2 淘汰;

3 号小朋友报数 3;

4 号小朋友报数 4 淘汰;

5 号小朋友报数 5;

1 号小朋友报数 6 淘汰;

3 号小朋友报数 7;

5 号小朋友报数 8 淘汰;

3 号小朋友获胜。

让用户给定 n 和 k, 请问最后获胜的小朋友编号为多少?

三. 实验步骤和结果

1. 第一题程序如下:

```
n=int(input('请输入整数 n: '))
for i in range(1,n+1):
    for j in range(n-i):
        print(' ',end='')           #输出空格
    for j in range(2*i-1):
        print('*',end='')           #输出*号
    print()
```

思路：每一行只需要分成两段：前面的空格和*号分别输出

运行结果如下：

```
请输入整数n: 1      请输入整数n: 3      请输入整数n: 5  
*                    *                    *  
  
***                  ***  
  
*****              *****  
  
                      *****  
                      *****
```

程序如下:

2. 第二题程序如下:

```
n=int(input())
a=[[None]*(n+1) for _ in range(n+1)] #开拓二维列表
for i in range(1,n+1): #初始化第一行和第一列
    a[0][i]=i
    a[i][0]=i
for i in range(1,n+1): #填上相乘所得数字
    for j in range(1,i+1):
        a[i][j]=i*j
for i in range(n+1): #按要求格式输出
    for j in range(n+1):
        if a[i][j]:
            print(a[i][j],end='\t') #按 tab 格式要求输出
        else:
            print(' ',end='\t')
    print()
```

思路：将需要输出的数先分类存进二维列表中，再格式化输出

运行结果如下:

```

请输入整数n: 9
          1   2   3   4   5   6   7   8   9
1         1
2        2   4
3       3   6   9
4      4   8  12  16
5     5  10  15  20  25
6    6  12  18  24  30  36
7    7  14  21  28  35  42  49
8    8  16  24  32  40  48  56  64
9    9  18  27  36  45  54  63  72  81
    
```

3. 第三题程序如下:

```
n=int(input('请输入整数 n: '))
for i in range (1,n+1):    #上半部分
    #输出一行前面的空格
    for j in range(n-i):
        print(' ',end='')
    #输出*号
    for j in range(n+i-1):
        print('*',end=' ')
    print()

for i in range(n-1,0,-1):  #下半部分 (只需改变第一行的 i 循环即可)
    # 输出一行前面的空格
    for j in range(n - i):
        print(' ', end='')
    # 输出*号
    for j in range(n + i - 1):
        print('*', end=' ')
    print()
```

思路: 可将图形分为两部分, 又因为上下对称, 所以只需要编写一半的图形
另一半倒序循环一遍即可

运行结果如下:

```
请输入整数n: 3      请输入整数n: 4

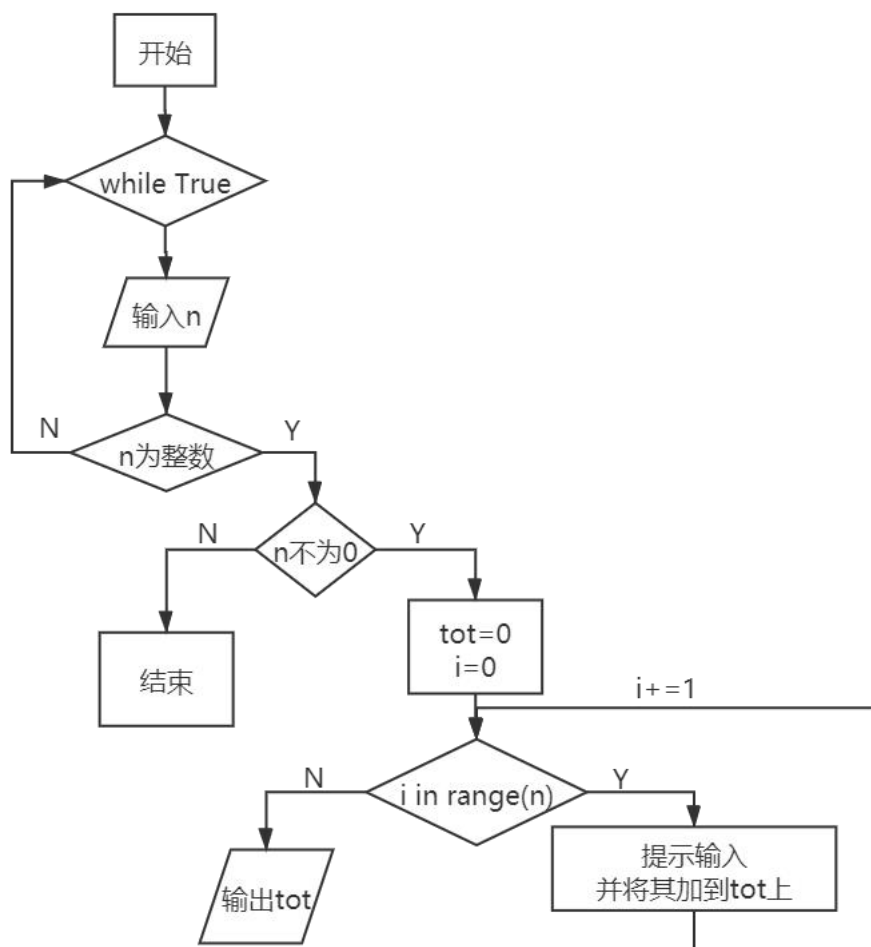
  *   *   *
 *   *   *   *
*   *   *   *   *
 *   *   *   *
  *   *   *

      *   *   *   *
      *   *   *   *   *
      *   *   *   *   *   *
      *   *   *   *   *   *
      *   *   *   *   *
      *   *   *   *   *
```

4. 第四题程序如下:

```
while 1:
    n=input('Please input the number of numbers: ')
    if n.isdigit():    #判断输入是否合法
        break
    else:
        print('当前的输入非法需要重新输入数值')
n=int(n)    #while 循环结束后, n 一定是整数
if n!=0:    #如果输入 'n=0' 代表退出程序
    tot=0
    for i in range(n):
        tot+=int(input('Please input number %d:'%(i+1)))#格式化
    print('sum=',tot)
```

- 思路：（1）用 **while** 循环读入，直到合法
（2）判断 **n** 读入的合法 **n** 是否为 0
（3）若不为 0 则继续读入，并加至 **tot** 总和



运行结果如下：

(1) Please input the number of numbers: *dsad*

当前的输入非法需要重新输入数值

Please input the number of numbers: *[1,2]*

当前的输入非法需要重新输入数值

Please input the number of numbers: *0*

Process finished with exit code 0

(2) Please input the number of numbers: *dasda*

当前的输入非法需要重新输入数值

Please input the number of numbers: *3*

Please input number 1: *5*

Please input number 2: *6*

Please input number 3: *7*

sum= 18

5. 第五题程序如下:

```
#判断素数的函数
def is_prime(n):
    if n==1:
        return False
    for i in range(2,int(n**0.5)+1):
        if n%i==0:
            return False
    return True
#主程序
n=int(input('请输入整数 n: '))
a=[]
for i in range(1,n):
    if is_prime(i):
        a.append(i)
for i in range(1,len(a)-1):
    print(a[i-1],end=', ') #因格式要求将最后一个素数单独输出
#防止没有素数时程序报错
if a!=[]:
    print(a[len(a)-1])
```

思路: (1) 根据区间依次枚举其中的整数
(2) 用 `is_prime` 函数判断枚举到的数是否为素数
(3) 由于格式要求用列表先将找到的素数存下来

运行结果如下:

(1) 请输入整数n: 2

Process finished with exit code 0

(2) 请输入整数n: 11

2, 3, 7

(3) 请输入整数n: 20

2, 3, 5, 7, 11, 13, 19

6. 第六题程序如下:

```
a=int(input('请输入整数 a: '))
n=int(input('请输入整数 n: '))
tot=0
aa=a
for i in range(n):#共有 n 个数相加, 只需控制循环次数即可
    tot+=aa
    aa=aa*10+a
print(tot)
```

思路：用 **aa** 作为每次循环时需要加上的数，**a** 保留用来计算下一次的 **aa**

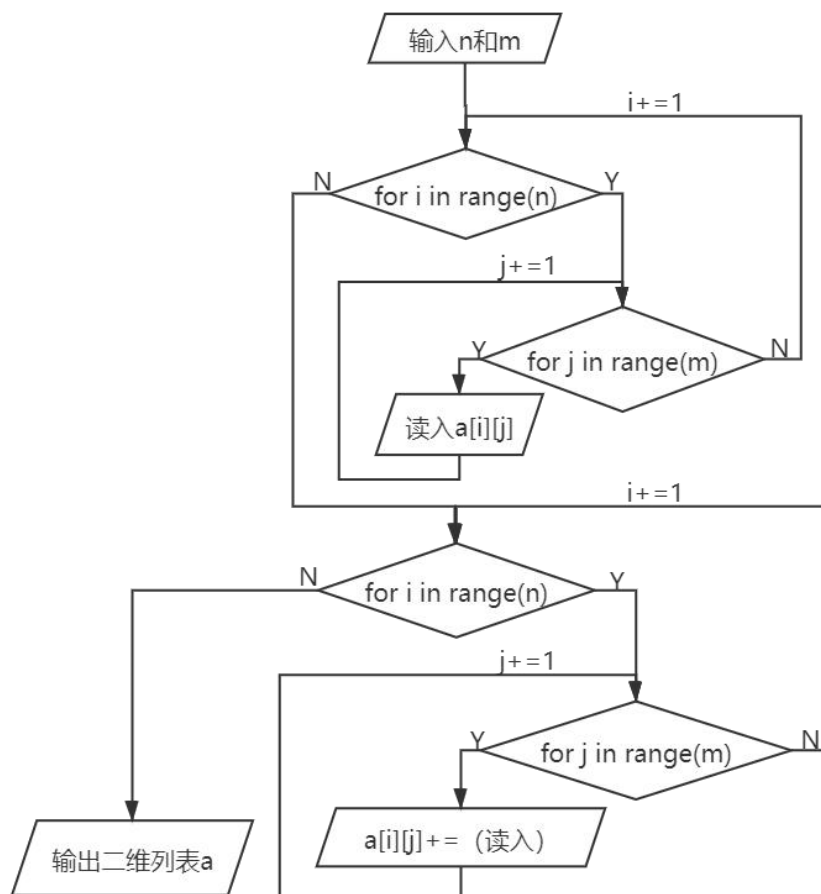
运行结果如下：

| | |
|---------------|---------------|
| (1) 请输入整数a: 2 | (2) 请输入整数a: 1 |
| 请输入整数n: 4 | 请输入整数n: 9 |
| 2468 | 123456789 |

7. 第七题程序如下：

```
n=int(input('Please input the number of rows: '))
m=int(input('Please input the number of columns: '))
a=[[]for _ in range(m)]for _ in range(n)] #开拓二维列表
#读入 A
for i in range(n):
    for j in range(m):
        a[i][j]=int(input('Please input A[{:0},{:1}]: '.format(i,j)))
#直接计算 C
for i in range(n):
    for j in range(m):
        a[i][j]+=int(input('Please input B[{:0},{:1}]: '.format(i,j)))
print('C=',a)
```

思路：（1）在列表中使用 for 语句来开二维列表
（2）因为本题中只需要用一次 **B**，所以可以不创建 **B** 和 **C** 矩阵，以节省空间
（3）使用格式化来输出读入提示中变化的行和列



运行结果如下：

```
(1) Please input the number of rows: 2
    Please input the number of columns: 3
    Please input A[0,0]: 1
    Please input A[0,1]: 2
    Please input A[0,2]: 3
    Please input A[1,0]: 4
    Please input A[1,1]: 5
    Please input A[1,2]: 6
    Please input B[0,0]: 6
    Please input B[0,1]: 5
    Please input B[0,2]: 4
    Please input B[1,0]: 3
    Please input B[1,1]: 2
    Please input B[1,2]: 1
    C= [[7, 7, 7], [7, 7, 7]]
```

```
(2) Please input the number of rows: 3
    Please input the number of columns: 2
    Please input A[0,0]: 1
    Please input A[0,1]: 2
    Please input A[1,0]: 1
    Please input A[1,1]: 2
    Please input A[2,0]: 1
    Please input A[2,1]: 2
    Please input B[0,0]: 1
    Please input B[0,1]: 2
    Please input B[1,0]: 1
    Please input B[1,1]: 2
    Please input B[2,0]: 1
    Please input B[2,1]: 2
    C= [[2, 4], [2, 4], [2, 4]]
```


8. 第八题程序如下:

```
n=int(input('请输入整数 n: '))
k=int(input('请输入整数 k: '))

b=[0 for _ in range(n+1)] #b 标记列表, 表示小朋友是否淘汰
x=0 #当前是编号为 x 的小朋友
step=0 #报数已经报到了 step 号
out_num=0 #已经淘汰的小朋友数量

#模拟游戏开始:
while True:
    x=x+1 if x!=n else 1
    if b[x]==0: #编号为 x 的小朋友没有被淘汰:
        step+=1
        if step%k==0: #此时淘汰
            out_num+=1
            b[x]=1
        if out_num==n-1: #判断是否已经决出了胜利者
            break

#寻找最终赢家:
for i in range(1,n+1):
    if b[i]==0:
        print('%d 号小朋友获胜'%i)
        break
```

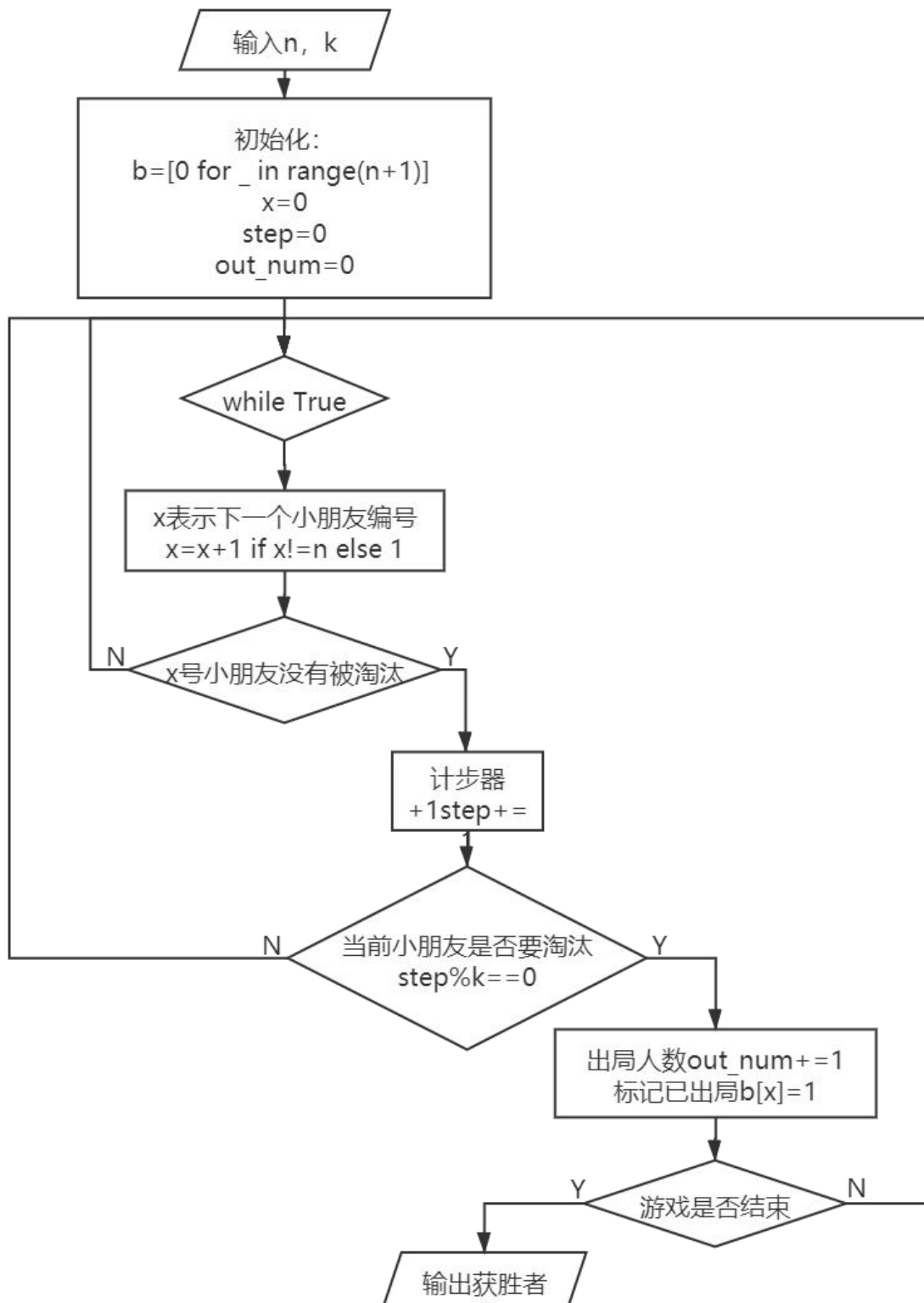
思路: (1) 采用标记数组, 下标对应小朋友编号, 假如淘汰将其值改变为 1
(2) 用 step 表示有效步数, 无效步数对应的就是已经淘汰的小朋友
(3) 采用 while 循环, 并结合 out_num 来判断是否决出了胜利者

运行结果如下:

| | | |
|-----------|-----------|------------|
| 请输入整数n: 5 | 请输入整数n: 3 | 请输入整数n: 10 |
| 请输入整数k: 2 | 请输入整数k: 3 | 请输入整数k: 3 |
| 3号小朋友获胜 | 2号小朋友获胜 | 4号小朋友获胜 |

流程图如下:

流程图如下：



拓展题:

1. 猜数字

程序如下:

```
import random
x=random.randint(100,999) #生成一个三位随机数
n=0
tot=0
while n!=x: #终止条件
    n=int(input('请输入所猜之数: '))
    if n<x:
        print('小了')
    if n>x:
        print('大了')
    tot+=1
print('对了')
print('您一共猜了%d 次'%tot) #格式化输出
```

思路: (1) 生成一个随机三位数

(2) 用 while 循环来不断提示输入, 然后比较输出提示, 直到猜对

(3) 格式化输出猜对所用次数

(4) 由于 1 到 10000 中的随机数需要猜的次数较多, 所以将范围缩小至三位数

运行结果如下:

请输入所猜之数: 500
大了
请输入所猜之数: 250
大了
请输入所猜之数: 175
大了
请输入所猜之数: 150
大了
请输入所猜之数: 130
大了
请输入所猜之数: 110
小了
请输入所猜之数: 120
小了
请输入所猜之数: 125
大了
请输入所猜之数: 123
对了
您一共猜了9次

请输入所猜之数: 500
小了
请输入所猜之数: 750
小了
请输入所猜之数: 800
小了
请输入所猜之数: 900
小了
请输入所猜之数: 950
小了
请输入所猜之数: 987
大了
请输入所猜之数: 970
小了
请输入所猜之数: 980
小了
请输入所猜之数: 985
大了
请输入所猜之数: 984
大了
请输入所猜之数: 983
对了
您一共猜了11次

2. 矩阵乘法

程序如下:

```
n1=int(input('n1:'))
m1=int(input('m1:'))
n2=int(input('n2:'))
m2=int(input('m2:'))
#开辟二维列表
a=[[None for _ in range(m1)]for _ in range(n1)]
b=[[None for _ in range(m2)]for _ in range(n2)]
c=[[0 for _ in range(m2)]for _ in range(n1)]
#读入矩阵 a
for i in range(n1):
    for j in range(m1):
        a[i][j]=int(input('A({:0},{:1}):'.format(i,j)))
#读入矩阵 b
for i in range(n2):
    for j in range(m2):
        b[i][j]=int(input('B({:0},{:1}):'.format(i,j)))
#计算 c
for i in range(n1):
    for j in range(m2):
        for k in range(m1):
            c[i][j]=c[i][j]+(a[i][k]*b[k][j])

print(c)
```

思路: (1) 用在外层用两层循环枚举 c 的行和列
(2) 根据公式, 里层用 k 枚举 a 的列和 b 的行

运行结果如下:

```
n1:2
m1:2
n2:2
m2:1
A(0,0):5
A(0,1):4
A(1,0):1
A(1,1):2
B(0,0):3
B(1,0):2
[[23], [7]]
```

```
n1:2
m1:3
n2:3
m2:2
A(0,0):4
A(0,1):5
A(0,2):2
A(1,0):3
A(1,1):6
A(1,2):1
B(0,0):2
B(0,1):1
B(1,0):3
B(1,1):6
B(2,0):5
B(2,1):4
[[33, 42], [29, 43]]
```

3. 输出[100,1000)以内的全部完数(因子之和等于自己, 如 $6=1+2+3$)

程序如下:

#func 函数来判断输入它是否是完数

```
def func(n):
    tot=0
    for i in range(1,n):
        if n%i==0:
            tot+=i
    if n==tot:
        return True
    else:
        return False
#按题目要求进行枚举
for i in range(100,1001,1):
    if func(i):
        print(i,end=' ')
```

思路: (1) 枚举所求范围内所有数
(2) 判断是否为完数

运行结果如下:

496

四. 实验总结 (包括对老师的建议)

(1) 掌握了循环结构程序设计方法

掌握了 continue 和 break 语句

掌握了 for 语句和 while 语句以及嵌套结构的使用方法

逐步建立起了组织复杂程序逻辑和流程的能力

(2) Python 里缩进是四个空格不是一个 tab, tab 之所以能在 pycharm 中可以用于缩进, 是因为 pycharm 自动将 tab 改成了四个空格

(3) 程序结构复杂时可以用函数, 注释和 pass 语句来使程序结构清晰

(4) While 条件较难写出时, 可以用 if 和 break 结合来结束循环