

## 实验报告 4

WA2214014 杨跃浙 人工智能 2 班

### 实验内容:

#### 实验内容 1-函数设计与使用

- 1、假设一段楼梯共 15 个台阶，小明一步最多能上 3 个台阶。编写程序计算小明上这段楼梯一共有多少种方法。要求给出递推法和递归法两种代码。
- 2、尼姆游戏：两个玩家轮流从一堆物品中拿走一部分。在每一步中,玩家至少拿走一个并且最多只能拿走一半物品,然后轮到下一个玩家。拿走最后一个物品的玩家输掉游戏。在聪明模式中,计算机每次拿走一定数量的物品使得堆的大小是 2 的幂次方减 1，也就是 3、7、15、31、63 等。如果有一定数量的剩余物品，计算机就随机拿走一些。编写程序,模拟聪明版本的尼姆游戏。
- 3、有  $n$  个人围成一圈，从 1 开始按顺序编号，从第一个人开始从 1 到(假设  $k=3$ )报数，报到  $k$  的人退出圈子；然后圈子缩小，从下一个人继续游戏，问最后留下的是原来的第几号。编写程序，模拟上面的游戏，要求初始人数  $n$  和报数临界值  $k$  可以自由指定。运行程序并观察游戏进行的过程。使用两种方法实现，并简单分析其优劣。
- 4、某商场用轮盘抽奖，轮盘的 $[0^\circ, 30^\circ)$ 对应一等奖， $[30^\circ, 108^\circ)$ 对应二等奖， $[108^\circ, 360^\circ)$ 对应三等奖。使用 0~360 的随机数表示消费者转动轮盘后指针所处的位置。编写程序，模拟该游戏，并试玩 10000 次，记录每个奖项的中奖次数。
- 5、有 A,B,C 3 个底座，A 座上有 64 个盘子，且大小不等，**所有底座上要求：大的在下,小的在上**。把这 64 个盘子从 A 座移到 C 座，可以利用 B 座，每次只能允许移动一个盘子。编写函数,接收一个表示盘子数量的参数和分别表示源、目标、临时底座的参数，然后输出详细移动步骤和每次移动后 3 个底座上的盘子分布情况。
- 6、抓狐狸小游戏。假设共有 5 个洞口，小狐狸在其中一个洞口,然后玩家随机打开一个洞口。抓到则成功，失败则重新抓。但是狐狸会在玩家重抓之前跳到隔壁洞口里。如果在规定的次数内抓到了狐狸则游戏成功；否则失败。
- 7、假如你在参加一个游戏节目，前方有 3 道门可以选择，其中一个门后面是汽车，另外两个门后面是山羊。主持人知道每个门后面是什么。你选择一个门，例如 1 号门，这时主持人打开了另一个门（例如 3 号门，后面是一只山羊），并问你：“你想改选 2 号门吗？”，然后根据你的选择确定最终要打开的门，并确定你获得山羊(输)或者汽车(赢)。编写程序，模拟上面的游戏。

#### 实验内容 2-函数与正则表达式

- 1、编写程序，生成 200 个人的模拟信息，包括姓名、性别、年龄、电话号码、家庭住址、电子邮箱地址，把生成的信息写入文本文件，每行存放一个人的信息。

```
def main(filename):  
    with codecs.open(filename, 'w', 'utf-8') as fp:  
        fp.write('Name,Sex,Age,TelNO,Address,Email\n')  
        #随机生成 200 个人的信息  
        for i in range(200):  
            #生成信息
```

```
line = ','.join([name, sex, age, tel, address, email]) + '\n'
fp.write(line)
```

2、利用**正则表达式**在给定的文本中查找 AABB 形式的语句。

行尸走肉、金蝉脱壳、百里挑一、金玉满堂、背水一战、霸王别姬、天上人间、不吐不快、海阔天空、情非得已、满腹经纶、兵临城下、春暖花开、插翅难逃、黄道吉日、天下无双、偷天换日、两小无猜、卧虎藏龙、珠光宝气、簪缨世族、花花公子、绘声绘影、国色天香、相亲相爱、八仙过海、金玉良缘、掌上明珠、皆大欢喜、浩浩荡荡、平平安安、秀秀气气、斯斯文文、高高兴兴

3、利用**正则表达式**，查找字符串中每个字符的首次出现。查找字符串中每个字符的首次出现。给定一个任意字符串，要求得到一个新字符串，重复字符只保留一个，并且新字符串中的字符保持在原字符串中首次出现的先后顺序。例如，abcdaaabe 处理后应得到 abcde。

**实验原理：**

## 1. 主要代码

### 实验内容 1-函数设计与使用

```
def test1():
    def Recursive(n, max_steps):
        if n <= 0:
            return 1
        if n == 1:
            return 1
        dp = [0] * (n + 1)
        dp[0] = 1
        dp[1] = 1
        for i in range(2, n + 1):
            for j in range(1, max_steps + 1):
                if i - j >= 0:
                    dp[i] += dp[i - j]
        return dp[n]

    def Induction(n, max_steps):
        if n <= 0:
            return 1
        if n == 1:
            return 1
        if n < max_steps:
            return Induction(n - 1, max_steps)
        return Induction(n - 1, max_steps) + Induction(n - max_steps, max_steps)

    n = 15
    max_steps = 3
    result_R = Recursive(n, max_steps)
    result_I = Induction(n, max_steps)
    print("小明上这段楼梯一共有{}种方法(递推)。\\n 小明上这段楼梯一共有{}种方法(递归)。".format(result_R, result_I))

def test2():

    import random
    import math
```

```

def taken(num):
    if int(math.log2(num+1))==math.log2(num+1) : return False
    return True
def nimu_game(choice,items):
    while items > 0:
        if choice=='y':
            while True:
                take=int(input("请你输入你想拿走几个物品："))
                if ((take<1) or (take>int(items/2))) and (items!=1):
                    print("玩家至少拿走一个并且最多只能拿走一半物品")
                else:
                    break
            items=items-take
            print("你拿走了", take, "个物品，剩余", items, "个物品。")
            if items==0:
                print("你输了！")
                break
            if taken(items) or (items==1):
                take=items-2**int(math.log2(items))+1
                items=items-take
                print("电脑拿走了", take, "个物品，剩余", items, "个物品。")
                if items==0:
                    print("你赢了！")
                    break
            else:
                take=random.randint(1,items//2)
                print("电脑拿走了", take, "个物品，剩余", items, "个物品。")
        if choice=='n':
            if taken(items) or (items==1):
                take=items-2**int(math.log2(items))+1
                items=items-take
                print("电脑拿走了", take, "个物品，剩余", items, "个物品。")
                if items==0:
                    print("你赢了！")
                    break
            else:
                take=random.randint(1,items//2)
                print("电脑拿走了", take, "个物品，剩余", items, "个物品。")
        while True:
            take=int(input("请你输入你想拿走几个物品："))
            if ((take<1) or (take>int(items/2)))and (items!=1):
                print("玩家至少拿走一个并且最多只能拿走一半物品")
            else:
                break
        items=items-take
        print("你拿走了", take, "个物品，剩余", items, "个物品。")
        if items==0:
            print("你输了！")
            break

s=input("请你选择先手拿或后手拿\ny 代表选择先手\nn 代表选择后手\n(y/n):")
item=int(input("总共的物品数量："))

```

```
nimu_game(s,item)
```

```
def test3():
```

```
def game_recursive(n, k):  
    if n == 1:  
        return 1  
    else:  
        return (game_recursive(n - 1, k) + k - 1) % n + 1
```

```
def game_loop(n, k):  
    people = list(range(1, n + 1))  
    index = 0  
    while len(people) > 1:  
        index = (index + k - 1) % len(people)  
        del people[index]  
    return people[0]
```

```
n = int(input("请输入初始人数: "))  
k = int(input("请输入报数临界值: "))  
result_recursive = game_recursive(n, k)  
result_loop = game_loop(n, k)  
print("最后留下的是原来第{}号 (递归)。\\n 最后留下的是原来第{}号 (循环)。"  
      ".format(result_recursive,result_loop))
```

```
def test4():
```

```
import random
```

```
def get_random():  
    return random.randint(0, 360)
```

```
def get_prize(angle):  
    if 0 <= angle < 30:  
        return "一等奖"  
    elif 30 <= angle < 108:  
        return "二等奖"  
    else:  
        return "三等奖"
```

```
prize_count = {"一等奖": 0, "二等奖": 0, "三等奖": 0}
```

```
for i in range(10000):  
    angle = get_random()  
    prize = get_prize(angle)  
    prize_count[prize] += 1
```

```
print("一等奖中奖次数: ", prize_count["一等奖"])  
print("二等奖中奖次数: ", prize_count["二等奖"])  
print("三等奖中奖次数: ", prize_count["三等奖"])
```

```
def test5():
    def hanoi(n, source, target, temp):
        if n == 1:
            print(f"将盘子从 {source} 移动到 {target}")
            return
        hanoi(n - 1, source, temp, target)
        print(f"将盘子从 {source} 移动到 {target}")
        hanoi(n - 1, temp, target, source)

    hanoi(4, 'A', 'C', 'B')
```

```
def test6():

    import random

    def catch():
        hole = random.randint(1, 5)
        attempts = 0
        max_attempts = 5

        while attempts < max_attempts:
            player = int(input("请输入一个洞口 (1-5) : "))
            print("你总共有 5 次尝试机会")
            attempts += 1

            if player == hole:
                print("恭喜你抓到了狐狸!")
                return
            else:
                print("很遗憾，你没有抓到狐狸。")
                if attempts < max_attempts:
                    path = random.choice([-1, 1])
                    if hole == 5: path = -1
                    if hole == 1: path = 1
                    hole = hole + path
                    print("狐狸跳到隔壁洞口里了，请继续尝试。")
                else:
                    print("你没有在规定次数内抓到狐狸。游戏失败。")

    catch()
```

```
def test7():

    import random

    def game():
        doors = [1, 2, 3]
        car_door = random.choice(doors)
        goat_doors = [door for door in doors if door != car_door]

        chosen_door = random.choice(doors)
        opened_door = [door for door in doors if door != chosen_door and door != goat_doors[0]][0]
```

```

print("你选择了门", chosen_door)
print("主持人打开了门", opened_door, "后面是一只山羊")

if input("你想改选门吗? (y/n): ") == 'y':
    chosen_door = int(input("请输入你想要选择的门号: "))

if chosen_door == car_door:
    return "你赢得了车! "
else:
    return "你输了! "

print(game())

if __name__ == '__main__':
    test1()
    test2()
    test3()
    test4()
    test5()
    test6()
    test7()

```

## 实验内容 2-函数与正则表达式

*# -\*- coding: utf-8 -\*-*

```

def test1():

    import random
    import string
    import codecs

    def randomstring(length):
        letters = string.ascii_lowercase
        return ''.join(random.choice(letters) for i in range(length))

    def randomtel():
        return '1' + ''.join(random.choice('0123456789') for i in range(9))

    def randomemail():
        return randomstring(5) + '@' + randomstring(5) + '.com'

    def main(filename):
        with codecs.open(filename, 'w', 'utf-8') as fp:
            fp.write('Name,Sex,Age,TelNO,Address,Email\n')
            # 随机生成 200 个人的信息
            for i in range(200):
                # 生成信息
                name = randomstring(5)
                sex = random.choice(['男', '女'])
                age = str(random.randint(18, 60))

```

```

        tel = randomtel()
        address = randomstring(10)
        email = randomemail()
        line = ','.join([name, sex, age, tel, address, email]) + '\n'
        fp.write(line)

    main('people_info.txt')

def test2():
    import re

    text = "行尸走肉、金蝉脱壳、百里挑一、金玉满堂、背水一战、霸王别姬、天上人间、不吐不快、海阔天空、情非得已、满腹经纶、兵临城下、春暖花开、插翅难逃、黄道吉日、天下无双、偷天换日、两小无猜、卧虎藏龙、珠光宝气、簪缨世族、花花公子、绘声绘影、国色天香、相亲相爱、八仙过海、金玉良缘、掌上明珠、皆大欢喜、浩浩荡荡、平平安安、秀秀气气、斯斯文文、高高兴兴"
    pattern = r'(\.)2(\.)3'
    result=[]
    for text in text.split(','):
        if re.match(pattern, text): result.append(text)
    print(result)

def test3():

    import re

    def remove(s):
        return ".join(re.findall(r'\w(?!.*\1)', s[::-1]))[::-1]"

    s = input()
    result = remove(s)
    print(result)

if __name__=='__main__':
    test1()
    test2()
    test3()

```

## 2. 运行结果

### 实验内容 1-函数设计与使用

C:\Users\yangy\AppData\Local\Programs\Python\Python38\python.exe E:/Python/Project/Test/Test4/Test1.py

小明上这段楼梯一共有5768种方法(递推)。

小明上这段楼梯一共有5768种方法(递归)。

请你选择先手拿或后手拿

y代表选择先手

n代表选择后手

(y/n):y

总共的物品数量: 100

请输入你想拿走几个物品: 45

你拿走了 45 个物品, 剩余 55 个物品。

电脑拿走了 24 个物品, 剩余 31 个物品。

请输入你想拿走几个物品: 20

玩家至少拿走一个并且最多只能拿走一半物品

请输入你想拿走几个物品: 13

你拿走了 13 个物品, 剩余 18 个物品。

电脑拿走了 3 个物品, 剩余 15 个物品。

请输入你想拿走几个物品: 4

你拿走了 4 个物品, 剩余 11 个物品。

电脑拿走了 4 个物品, 剩余 7 个物品。

请输入你想拿走几个物品: 3

你拿走了 3 个物品, 剩余 4 个物品。

电脑拿走了 1 个物品, 剩余 3 个物品。

请输入你想拿走几个物品: 1

你拿走了 1 个物品, 剩余 2 个物品。

电脑拿走了 1 个物品, 剩余 1 个物品。

请输入你想拿走几个物品: 1

你拿走了 1 个物品, 剩余 0 个物品。

你输了!

请输入初始人数: 10

请输入报数临界值: 3

最后留下的是原来第4号(递归)。

最后留下的是原来第4号(循环)。

一等奖中奖次数: 848

二等奖中奖次数: 2193

三等奖中奖次数: 6959

将盘子从 A 移动到 B

将盘子从 A 移动到 C

将盘子从 B 移动到 C

将盘子从 A 移动到 B

将盘子从 C 移动到 A

将盘子从 C 移动到 B

将盘子从 A 移动到 B

将盘子从 A 移动到 C

将盘子从 B 移动到 C

将盘子从 B 移动到 A

将盘子从 C 移动到 A

将盘子从 B 移动到 C

将盘子从 A 移动到 B

将盘子从 A 移动到 C

将盘子从 B 移动到 C

请输入一个洞口(1-5): 3

你总共有5次尝试机会

很遗憾, 你没有抓到狐狸。

狐狸跳到隔壁洞口里了, 请继续尝试。

请输入一个洞口(1-5): 3

你总共有5次尝试机会

很遗憾, 你没有抓到狐狸。

狐狸跳到隔壁洞口里了, 请继续尝试。

请输入一个洞口(1-5): 3

你总共有5次尝试机会

很遗憾, 你没有抓到狐狸。

狐狸跳到隔壁洞口里了, 请继续尝试。

请输入一个洞口(1-5): 3

你总共有5次尝试机会

恭喜你抓到了狐狸!

你选择了门 1

主持人打开了门 3 后面是一只山羊

你想改选门吗? (y/n): n

你赢得了车!

进程已结束, 退出代码0



## 实验内容 2-函数与正则表达式

```
C:\Users\yangy\AppData\Local\Programs\Python\Python38\python.exe E:/Python/Project/Test/Test4/Test2.py
['浩浩荡荡', '平平安安', '秀秀气气', '斯斯文文', '高高兴兴']
abcdaabe
abcde

进程已结束,退出代码0
```

### people\_info.txt 文件

```
Name,Sex,Age,TelNO,Address,Email
utmby,女,49,1505382686,fafbwtymyy,ubmkt@zuqmo.com
faqiz,女,34,1056173483,fsnyqdyrmi,gssjz@zgbux.com
qctez,女,27,1624616745,kinuwttute,uactu@mjfoo.com
qffqv,女,49,1404567467,dyfulydram,ujtzt@yswei.com
qelon,男,39,1246416364,besxanpaay,npqbl@sbhgm.com
nuivr,女,27,1572665556,owmzgziobr,vpkbl@qgmxd.com
dnvao,女,57,1910070054,xjmazjpuoq,ytzfo@rjcks.com
epewf,女,55,1120942811,vqmhygytrjw,mpxcf@zbsrz.com
fgyqp,女,48,1130854029,iikeiddxt,elmgk@gvjvn.com
ustnn,男,32,1402468664,nvlgsehvvh,ezyjs@dbnvy.com
zttvv,女,38,1620899295,abjiwkzffb,nqohg@fhyir.com
lkuka,男,54,1537480216,fdaexjhbfo,gzmhv@ddbbl.com
scias,女,48,1428998373,mfkexxmnut,iauqt@dmxys.com
bwojr,男,57,1478777164,xwlbggolxj,apemk@upzow.com
sckam,男,48,1694543981,fdofleqdpj,lungs@xdblc.com
jgdiv,男,28,1149915533,fyblaunqyx,ucsew@mrgnw.com
soxjy,男,28,1158439434,sylbytappw,augvm@uumxp.com
yymfd,女,47,1199021628,drtabqzezu,ctsob@gbkjb.com
pmluq,男,57,1423834341,xjyjhfjcgm,nkktk@okiwz.com
lpljw,女,49,1311962013,umsucowfso,rndrs@zspit.com
oomoc,女,36,1980254414,dkfdlyuapr,qjcej@sjsfuf.com
ewbhn,男,25,1346992296,ztanxgomcz,hcinf@zerpw.com
ormme,男,41,1335991383,hxcisrwsbj,xpjei@trnde.com
aelvg,男,51,1399125854,qcfpfhkgkj,pihjm@studa.com
ftuat,女,60,1204498399,tauieemkrf,pzgt@irbvr.com
ftdxu,女,47,1905460200,zcggjzcoxc,sukvt@myxej.com
kkxzt,男,25,1404316525,izryklogdg,aqzwm@brexo.com
ypxag,男,23,1104341087,pbwckpajoy,ypblv@upszp.com
kuwdb,女,39,1723395983,vzoetziprh,cwvvw@uewb.com
qaxxy,男,56,1276091978,mkdagqdafj,ijkgx@dnkz.com
vwmeo,女,39,1825074669,rjrjesksvt,sdkgx@iquzo.com
yrbai,男,48,1539044261,hfsedkahbm,ynrxe@ksegy.com
knmnj,女,25,1460860333,pkfuyjrsnq,jvfun@zltmu.com
gxdai,男,51,1549028961,wmukwmuqjr,ebric@gzwrn.com
```

## 小结与讨论:

### 实验内容 1-函数设计与表示

题 1: 爬楼梯问题。递推法使用动态规划的方法, 从底部向上计算每个台阶的方法数。递归法则是直接递归计算每个台阶的方法数。两种方法的时间复杂度都是  $O(n)$ , 空间复杂度都是  $O(n)$ 。

题 2: 尼姆游戏。尼姆游戏的必胜策略是在有  $2^n-1$  时选择后手, 其余情况选择先手, 所以在程序设计过程中增加了让玩家选择先手后手以及物品数的功能。并且对于先手和后手进行了不同的程序设计。同时为了游戏能够正常进行, 在玩家输入非法拿去数量时进行了提示, 直至玩家输入合法。

题 3: 约瑟夫环问题。采用循环和递推两种不同的方式分别模拟约瑟夫环问题。两种方法时间和空间复杂度相似, 循环算法需要开一个临时数组模拟, 递推算法则直接采用栈模拟。

题 4 中先设置不同的角度对应的不同的奖, 然后通过产生随机的方式模拟不同的角度。

题 5, 汉诺塔问题。用递归的方式模拟, 题目中要求输出 64 个盘子的情况, 考虑到盘子的增加将会使移动步数指数级增长, 故在和老师交流后, 把盘子数量改为 4 个。

题 6 中, 抓狐狸游戏, 需要注意的是在狐狸移动过程中需要注意边界, 在第一个洞或者第五个洞时移动方向受到限制。

题 7, 三门问题, 需要注意的是主持人需要打开的门后面一定是山羊。按照题目要求模拟该游戏。

### 实验内容 2-函数与正则表达式

题 1 中编写对应的函数生成相关的资料, 然后通过调用 `main()` 函数用 `with` 方法打开文件并写入信息, 最后保存在文件 `people_info.txt` 文件中。

题 2 中用正则表达式找出“AABB”形式的成语。需要注意的是要找 AABB 形式的成语, 而不是连续出现两次的字符。

题 3 中需要用正则表达式匹配重复字符, 并且实现只保留一个的功能。