# 第三章作业

1. while 循环和 for 循环

2. 对于表达式‘表达式 1or 表达式 2’，如果表达式 1 为 True，无论表达式 2 为何值，这个表达式都为 True

3.

3.py:

```
x=input('请输入年份:')
x=int(x)
if (x%400==0)or(x%4==0 and x%10!=0):
    print('Yes')
else:
    print('No')
```

4.

4py:

```
from random import *
a=[randint(0,100) for i in range(50)]
for i in a[:]:
    if i%2!=0:
        a.remove(i)
print(a)
```

5.

5. py:

```
from random import *
a=[randint(0,100) for i in range(20)]
print(sorted(a[::2],reverse=True))
```

6.

6. py:

```
x=input('请输入一个不小于 1000 的整数:')
x=int(x)
print(x,'=',end='')
from math import *
while(0 in [x%i for i in range(2,int(sqrt(x))+1)]):
    for j in range(2,int(sqrt(x))+1):
        if x%j==0:
            print(j,'*',end='')
            x=x//j
            break
print(x)
```

7.

7. py:

```
x=0
for i in range(1,100,2):
    x+=i
print(x)
```

```
x=0
y=1
while(y<100):
    x+=y
    y+=2
print(x)
```

**8.**

8. py:

```
def my_list():
    list0 = [1, 2, 3, 4]
    list1 = []
    for i in list0:
        for j in list0:
            if i == j:
                continue
            for k in list0:
                if (i == k) or (j == k):
                    continue
                for l in list0:
                    if (i == l) or (j == l) or (k == l):
                        continue
                    result = 1000 * i + 100 * j + 10 * k + l
```

```
        list1.append(result)
    return list1


prime_list=[p for p in my_list() if 0 not in [p%d for d in range(2,int(p**0.5+1))]]
print(prime_list)
```

9.

9. py:

```
x=int(input('请输入 x 的值:'))
if x<0:
    y=0
elif 0<=x<5:
    y=x
elif 5<=x<10:
    y=3*x-5
elif 10<=x<20:
    y=0.5*x-2
else:
    y=0
print(y)
```

10.

```
from random import randrange
```

```
def init():
    '''返回一个字典，键为 3 个门号，值为门后面的物品'''
    result = {i: 'goat' for i in range(3)}
    r = randrange(3)
    #在某个随机的门后面放一辆汽车，其他两个门后面仍然是山羊
    result[r] = 'car'
    return result
def startGame():
    #获取本次游戏中每个门的情况
    doors = init()
    #获取玩家选择的门号
    while True:
        try:
            firstDoorNum = int(input('Choose a door to open:'))
            assert 0<= firstDoorNum <=2
            break
        except:
            print('Door number must be between {} and {}'.format(0, 2))

    #主持人查看另外两个门后的物品情况
    #字典的 keys()方法返回结果可以当作集合使用，支持使用减法计算差集
    for door in doors.keys()-{firstDoorNum}:
        #打开其中一个后面为山羊的门
        if doors[door] == 'goat':
            print('"goat" behind the door', door)
            #获取第三个门号，让玩家纠结
            thirdDoor = (doors.keys()-{door, firstDoorNum}).pop()
            change = input('Switch to {}?(y/n)'.format(thirdDoor))
            finalDoorNum = thirdDoor if change=='y' else firstDoorNum
            if doors[finalDoorNum] == 'goat':
                return 'I Win!'
            else:
                return 'You Win.'
while True:
    print('='*30)
    print(startGame())
    r = input('Do you want to try once more?(y/n)')
    if r == 'n':
        break
```

## 11.

```
import random
```

```
print('''
    欢迎来到尼姆的游戏!
    规则如下:
        将硬币分成几堆;
        两个人轮流取硬币;
        每次取硬币只能从同一堆中取出，枚数不限，但至少要取一枚;
        取走最后一枚硬币的是输家，逼迫对方取到最后一枚硬币是赢家。
''')

# 1.生成硬币堆 用一个字典来存储有多少硬币
while True:
    try:
        line = int(input("请输入行数:"))
        lineNum = int(input("请输入第一行有几个硬币:"))
        break
    except Exception:
        print("输入错误!请重新输入")


# 生成硬币堆
def generate_dict(line, lineNum):
    d = dict()
    for i in range(line):
        d[i] = lineNum + i

    return d


# 展示硬币堆
def show_dict(d):
    for i in range(len(d)):
        if d[i] > 0:
            print("第", i + 1, "行:", end="\t")
            for j in range(d[i]):
                print("●", end="\t")
            print()


# 修改硬币堆
def updata_dict(d, line, count):
    d[line] = d[line] - count

# 根据参数生成行和硬币数
d1 = generate_dict(line, lineNum)
```

```
# d1 = generate_dict(3, 3) # TestCode
show_dict(d1)

countGame = 0    # 游戏进行的次数  单数为玩家,双数为机器人
line = 0  # 取硬币的第几行
count = 0     # 取硬币多少个硬币

while True:
    if sum(d1.values()) <= 0:    # 如果所有的硬币都为零则游戏结束
        break
    while True:
        # 开始游戏    玩家拿走硬币
        try:
            line = int(input("请输入要拿走第几行的硬币:")) - 1
            count = int(input("请输入要拿走几个硬币:"))
            if count <= d1[line]:
                break
        except Exception:
            print("输入错误!请重新输入")

    updata_dict(d1, line, count) #  根据玩家的意愿修改取走硬币
    countGame += 1
    show_dict(d1)    # 展示剩下的硬币
    print("你拿走了{}行的{}枚硬币".format(line + 1, count))
    # 机器人拿走硬币
    if sum(d1.values()) <= 0:    # 如果没有硬币则直接结束游戏
        break
    elif d1[line] > 0:    # 如果当前列是否还有有硬币
        RobotCount = random.randint(1, d1[line])
    else: # 寻找有硬币的行
        for i in range(len(d1)):
            if d1[i] > 0:
                line = i

    # 机器人拿走的硬币
    RobotCount = random.randint(1, d1[line])

    updata_dict(d1, line, RobotCount)
    countGame += 1
    print("机器人拿走了{}行的{}枚硬币".format(line + 1, RobotCount))
    show_dict(d1)

#   # 判断游戏结果
if countGame % 2 == 1:
```

```
    print("你输了")
else:
    print("你赢了")
```

## 12.

```
def isUgly(num):
        if num < 1:
            return False
        while num % 2 == 0 or num % 3 == 0 or num % 5 == 0:
            if num % 2 == 0:
                num //= 2
            elif num % 3 == 0:
                num //= 3
            elif num % 5 == 0:
                num //= 5
        if num != 1 and num != 2 and num != 3 and num != 5:
            return False
        else:
            return True
for i in range(100):
    if isUgly(i)==True:
        print(i,' ')
```