

# 深圳大学实验报告

课程名称： 现代程序设计

实验项目名称： 类与对象

学院： 电子与信息工程学院

专业：

指导教师： 邹文斌

报告人：  学号：  班级：

实验时间： 2023 年 11 月 13 日

实验报告提交时间： 2023 年 11 月 17 日

教务部制

## 一、 实验要求

- a) 函数
- b) 算法
- c) 需上交实验报告、py 源程序。

## 二、 实验环境

Python IDLE, Pycharm 等

## 三、 实验内容

### 1. Rectangle 类

创建一个名为 Rectangle 的类，具有宽 width 和高 height 两个属性。

实现一个方法 area()，用于计算矩形的面积。

创建好 Rectangle 类后，执行以下实例化一个 Rectangle 对象，可打印矩形的面积。

```
jx = Rectangle(5, 3)
s = jx.area()
print("面积:", s)
```

程序输出样例：

面积: 15

### 2. Person 类

创建一个名为 Person 的类，具有 name 和 age 两个属性。

创建好 Person 类后，执行以下程序实例化一个 Person 对象，可打印其姓名和年龄。

```
student1 = Person("小荔", 19)
print("姓名:", student1.name)
print("年龄:", student1.age)
```

程序输出样例：

姓名: 小荔

年龄: 19

### 3. 扑克牌

有  $N(3 \leq N \leq 50)$  张反面朝上的扑克牌排成一排，每次可以翻转任意 3 张扑克牌(正面翻转后为反面，反面翻转后为正面，且不考虑正面花色和点数)，请问最少翻转几次可以将  $N$  张扑克牌全部翻转为正面。

例如：

$N=5$ , 最少翻转 3 次可以使 5 张扑克牌全部翻转为正面朝上。

初始状态：



第 1 次：将第 1、2、3 张扑克牌进行翻转，翻转后如下图所示；



第 2 次：将第 1、2、4 张扑克牌进行翻转，翻转后如下图所示；



第 3 次：将第 1、2、5 张扑克牌进行翻转，翻转后如下图所示。



**输入描述：**输入一个正整数  $N$  ( $3 \leq N \leq 50$ ), 表示扑克牌的数量

**输出描述：**输出一个整数，如果可以将  $N$  张反面朝上的扑克牌全部翻转为正面朝上，输出最少需要翻转的次数，否则输出 -1

**样例输入：**5

**样例输出：**3

#### 4. 数据操作 (A+挑战题)

有一组正整数数据，现对这组数据按照如下操作：

- 1) 从这组数中找出两个相邻且相同的数，删掉其中一个数，剩下的一个数加 1（例如：两个相邻的 6，变成一个 7）；
- 2) 重复操作第 1 步，直到这组数据中没有相邻且相同的数时，操作结束。

现给定  $N$  ( $1 \leq N \leq 2000$ ) 个正整数，表示这一组数，请问按照要求操作结束后，这组数据中最大的数是多少。

注意：不同的操作方式得到的最后结果不同，要求最后的结果是所有操作方式中最大的。

例如：

当  $N=6$ ，这组数为 1、2、2、2、3、4 时，

可获得最大结果的操作如下：

第一次操作：将这组数据中后两个相邻的 2，变成 3，此时这组数变为 1，2，3，3，4；

第二次操作：将这组数据中两个相邻的 3，变成 4，此时这组数变为 1，2，4，4；

第三次操作：将这组数据中两个相邻的 4，变成 5，此时这组数变为 1，2，5；

此时这组数据中没有相邻且相同的数，操作结束，最大的数是 5。

非最大结果的操作如下：

第一次操作：将这组数据中前两个相邻的 2，变成 3，此时这组数变为 1，3，2，3，4；

此时这组数据中没有相邻且相同的数，操作结束，最大的数是 4。

所以按照要求操作结束后，这组数据中可获得的最大数是 5。

#### 输入描述：

第一行输入一个正整数  $N$  ( $1 \leq N \leq 2000$ )

第二行输入  $N$  个正整数 ( $1 \leq \text{正整数} \leq 40$ )，相邻两个数之间以一个空格隔开

#### 输出描述：

输出一个正整数，表示所有操作方式中最大的结果

### 5. 机甲战士 (A+挑战题)

潘多拉星球上的基地指挥官准备派遣一名机甲战士外出执行任务，为了完成任务，机甲战士需要随身携带足够的生存装备。生存装备里存储着氧气和燃料，且每个装备有一定的重量。机甲战士完成任务所需要携带的氧气和燃料都有最低限量，绝对不能少于这个最低限量，一次任务可以携带多个生存装备备用。

当给出最低氧气限量值  $M$  ( $1 \leq M \leq 30$ ) 和最低燃料限量值  $N$  ( $1 \leq N \leq 80$ )，生存装备的数量  $K$  ( $1 \leq K \leq 1000$ ) 及每个生存装备中氧气值  $a$  ( $1 \leq a \leq 30$ )、燃料值  $b$  ( $1 \leq b \leq 80$ ) 及装备重量  $c$  ( $1 \leq c \leq 100$ )，请计算出机甲战士要完成任务所需要携带的生存装备的总重量最低是多少，并输出这个最低值。

例如：

$M=6, N=40, K=5$ ，5 个生存装备中的氧气值、燃料值及装备重量分别如下：

编号	氧气值	燃料值	重量
1	5	5	30
2	7	15	40
3	4	20	50
4	3	25	80
5	2	30	90

可以满足要求的携带方法有：

2 号和 4 号装备的组合 (总重量为 120)；

2 号和 5 号装备的组合 (总重量为 130)；

3 号和 4 号装备的组合 (总重量为 130)；

.....

其中，生存装备的总重量最低为 120。

#### 输入描述：

第一行输入两个正整数  $M$  ( $1 \leq M \leq 30$ ) 和  $N$  ( $1 \leq N \leq 80$ )，分别表示氧气和燃料的最低限量值

第二行输入一个正整数  $K$  ( $1 \leq K \leq 1000$ ) 表示生存装备的个数

第三行开始输入  $K$  行，每行包括 3 个正整数  $a, b, c$  ( $1 \leq a \leq 30, 1 \leq b \leq 80, 1 \leq c \leq 100$ )，分别表示生存装备中氧气值和燃料值及装备的重量

**输出描述:**

输出一个整数，为机甲战士完成工作所需的生存装备的总重量最低值；如果携带所有的生存装备仍满足不了最低氧气限量值或最低燃料限量值，那么输出-1。

**样例输入:**

```
6 40
5
5 5 30
7 15 40
4 20 50
3 25 80
2 30 90
```

**样例输出:**

```
120
```

#### 四、实验过程

**思路:**

（涉及到算法实现的实验需阐述算法的逻辑关系）

**一、Rectangle 类**

1. 创建一个名为 Rectangle 的类，具有宽 width 和高 height 两个属性。
2. 实现一个方法 area()，用于计算矩形的面积。
3. 实例化一个 Rectangle 对象，打印矩形的面积。

**二、Person 类**

1. 创建一个名为 Person 的类，定义 name 和 age 两个属性。
2. 实例化对象并打印结果。

**三、扑克牌**

1. 定义一个函数 flip 用于计算需要翻转的次数。
2. 对于数量小于 4 的扑克牌，无法实现翻转，返回-1。
3. 对于 5 张扑克牌，返回 3。
4. 对于 3 的倍数张扑克牌，返回 N/3；对于其他数量的扑克牌，次数为 N/2+2。
5. 获取输入，进行计算并打印结果。

**完整代码:**

(必须有详细的注释)

### 一、Rectangle 类 Rectangle.py

```
class Rectangle: #创建一个类

    def __init__(self,width,height): #设置宽和高两个属性

        self.width = width

        self.height = height


    def area(self): #方法 area() 用于计算矩形面积

        return self.width * self.height

#实例化对象，打印结果

jx = Rectangle(5,3)

s = jx.area()

print("面积:",s)
```

### 二、Person 类 Person.py

```
class Person: #创建一个 Person 类，定义 name 和 age 两个属性

    def __init__(self,name,age):

        self.name = name

        self.age = age


#实例化对象并打印

student1 = Person("小荔",19)

print("姓名:",student1.name)

print("年龄:",student1.age)
```

### 三、扑克牌 Poker.py

```
def flip(N):

    if N < 4:

        return -1


    if N == 5:
```

```

        return 3

    if N % 3 == 0:

        # 对于 3 的倍数张扑克牌，需要翻转的次数为 N//3

        return N // 3

    else:

        # 其他张数扑克牌，需要翻转的次数为 N//2 + 2

        return N // 2 + 2

N = int(input("请输入需要翻转的扑克牌数量:"))

result = flip(N)

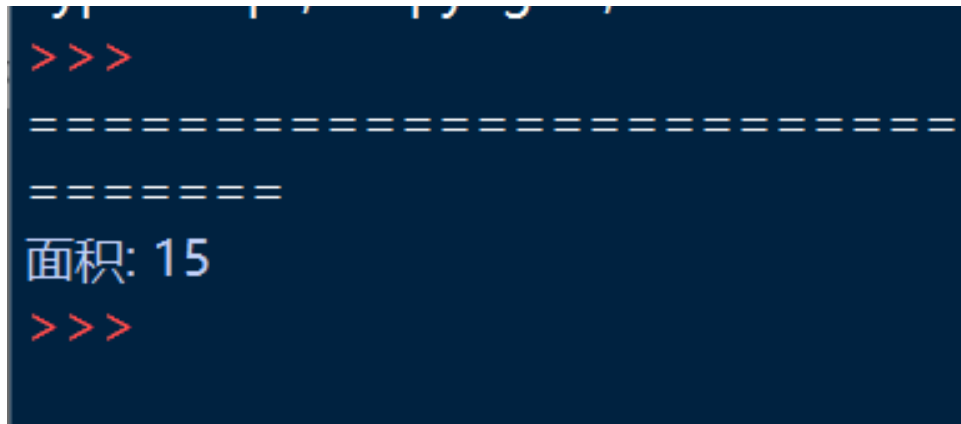
print("至少需要翻转的次数为:", result)

```

## 五、 实验结果

（运行结果，截图）

### 一、Rectangle 类



```

>>>
=====
面积: 15
>>>

```

### 二、Person 类

```
>>>
===== RES
=====
姓名: 小荔
年龄: 19
>>>
```

### 三、扑克牌

```
===== RE
=====
请输入需要翻转的扑克牌数量:5
至少需要翻转的次数为: 3
>>>
```

### 六、 实验心得

（本次实验遇到的问题，解决过程，有什么收获等）

- 1.简单认识了类并学习了其基本使用。
- 2.初步认识了面向对象编程的概念。



成绩评定:

实验过程（60 分）	实验结果（30 分）	心得体会（10 分）	总分（100 分）

指导教师签字：                      年    月    日

备注:

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。