

# 深圳大学实验报告

课程名称： 现代程序设计

实验项目名称： 程序流程控制

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 9 月 18 日

实验报告提交时间： 2023 年 9 月 21 日

教务部制

## 一、 实验要求

- a) 选择语句
- b) 循环控制
- c) 跳转控制
- d) 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. 编写程序对百分制成绩进行分档。 用户输入百分制分数，输出 ABCD 分档信息。90 分以上为 A，75~89 分为 B，60~74 分为 C，59 分以下为 D。
2. 编写一个程序，要求用户输入一个正整数 n，然后输出前 n 个素数。素数是只能被 1 和自身整除的大于 1 的整数。例如，当 n 为 6 时，输出前 6 个素数：2, 3, 5, 7, 11, 13。
3. 编写一个程序，要求用户输入一个正整数 n，然后输出一个 n 行的帕斯卡三角形。帕斯卡三角形是由数字组成的三角形，每个数字是由上方两个数字相加而得到的。例如，当 n 为 5 时，输出如下：

```
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
```

4. 求小数的某一位。分数  $a/b$  化为小数后，小数点后第 n 位的数字是多少？三个正整数 a, b, n 分三行输入， $0 < a < b < 100$ ， $1 \leq n \leq 10000$ 。输出数点后第 n 位的数字，注意 n 最大可达 10000。
5. 给定一个十进制正整数 N, 请从小到大输出  $1 \sim N$  之间(含 1 和 N)所有满足以下要求的数：
  1. 这个数转换为八进制后是一个回文数；
  2. 这个数是一个平方数。

例如：N=20, 在  $1 \sim 20$  之间满足要求的数有 1、4、9, 因为有，1 转换为八进制为 1, 是一个回文数；且  $1 = 1 \times 1$ , 是一个平方数；4 转换为八进制为 4, 是一个回文数；且  $4 = 2 \times 2$ , 是一个平方数；9 转换为八进制为 11, 是一个回文数；且  $9 = 3 \times 3$ , 是一个平方数。故输出 1 4 9。

## 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)

#### 任务 1:

1. 用户输入百分制分数, 将该百分制分数转换为数字型。
2. 根据题目要求, 利用条件判断语句 if, 对输入的分数进行判断并作出分档操作。

#### 任务 2:

1. 首先将用户输入的整数 n 转化为整型。
2. 设置一个计数器, 从 0 开始, 每找到一个素数计数器的值便+1, 直到 n。
3. 根据素数的定义, 设置素数判断从数字 2 开始。
4. 判断: 用数 num 对从 2 到 num 的平方根依次取模, 如果结果均为 0, 则说明 num 不是素数; 如果取模结果均不为 0, 则说明 num 为素数。
5. 每找到一个素数 num, 输出, 同时计数器 count 的值+1, 继续进入下一个循环。

#### 任务 3:

1. 首先获取行数, 将输入值转换为整数。
2. 考虑打印结果的格式, 计算三角形中最大的数字的宽度: 最后一行的行号是 n-1, 使用  $2^{n-1}$  来计算帕斯卡三角形中最大的数字, 便于后续结果的有序打印。
3. 设置外层循环 for i in range(n) 控制行数, 从第一行到第 n 行; 内层循环 for k in range(n-i-1) 打印每一行开头的空格, 用于使三角形居中对齐。
4. 内层循环 for j in range(i+1) 控制每行中的数字个数, 从第一列到第 i+1 列。打印当前的数字, 使用 .rjust(width) 右对齐并占据 width 的宽度;  $num = num * (i - j) // (j + 1)$  更新 num 的值, 通过计算组合数的方式得到下一个数字。
4. 每打印完一行, 用 print() 打印换行符, 进入下一行。

#### 任务 4:

1. 首先对三个正整数 a, b, n 分三行输入, 同时判断是否在题目规定范围内。
2. 定义一个空字符串 decimal 来储存小数部分的数字, 余数 remainder 为 a%b 的余数。
3. 用循环 for i in range(n) 来计算小数部分, 同时在每次循环中将余数乘以 10, 将商 (remainder//b) 的整数部分转换为字符串格式, 储存到 decimal 中, 同时更新这时的余数为 remainder%b 的余数。

4. 最后用字符串索引 `decimal[-1]` 打印出小数后第 `n` 位的数字。

#### 任务 5:

1. 首先获得整数 `N`，遍历  $1 \sim N$  的所有整数进行回文数判断。
2. 利用 `(num ** 0.5) ** 2 == num` 判断当前数字是否为平方数。
3. 利用 `oct(num)[2:]` 将数字转化为八进制，并去掉开头的标识，便于后续判断操作。
4. 使用切片操作 `num_oct[::-1]` 将 `num_oct` 字符串进行反转得到新字符串，利用的原理为：如果八一个进制数与其反转后的八进制数相等，则该八进制数为回文数。
5. 判断结束，输出范围内符合条件的回文数。

#### 完整代码:

（必须有详细的注释）

#### 任务 1: Grade\_grading.py

#用户输入百分制分数，并对输入的分数转换成数值型数字

```
score = eval ( input ("请输入百分制分数: "))
```

#对输入的分数范围进行判断，做出分档操作

```
if score >=90:
```

```
    print ("该分数分档为 A ")
```

```
elif 75<= score <=89:
```

```
    print ("该分数分档为 B ")
```

```
elif 60<= score <=74:
```

```
    print ("该分数分档为 C ")
```

```
else:
```

```
    print ("该分数分档为 D ")
```

#### 任务 2: Prime\_number.py

```
n = int(input("请输入一个正整数 n: "))
```

# 设置计数器，记录已找到的素数的数量

```
count = 0
```

# 从 2 开始判断

```
num = 2
```

```

while count < n:
    for i in range(2, int(num ** 0.5) + 1): # 遍历 2 到 num 的平方根之
间的数字
        if num % i == 0: # 如果 num 能够整除 i, 说明 num 不是素数
            break
    else:
        print(num) # 输出当前找到的素数
        count += 1 # 计数器加 1, 表示已找到一个素数
    num += 1 # 继续判断下一个数字

```

### 任务 3: Triangle.py

```

n = int(input("请输入一个正整数 n: "))

width = len(str(2**(n-1))) # 计算最大的数字的宽度

for i in range(n):
    # 在每行开头打印空格, 使三角形居中对齐
    for k in range(n - i - 1):
        print(" " * width, end=" ")
    num = 1
    # 打印当前行的数字
    for j in range(i + 1):
        print(str(num).rjust(width), end=" ")
        num = num * (i - j) // (j + 1)
    print()

```

### 任务 4: Decimal.py

```

a = int(input("请输入分子 a (0 < a < b < 100) : "))

```

```

b = int(input("请输入分母 b (0 < a < b < 100) : "))
n = int(input("请输入要求的小数点后第 n 位 (1 <= n <= 10000) : "))

if 0<a<b<100 and 1<=n<= 10000:
    # 使用长除法计算小数部分
    decimal = ""
    remainder = a % b

    for i in range(n):
        remainder *= 10 # 将余数乘以 10，相当于将小数点向右移动一位
        decimal += str(remainder // b) # 将商（余数除以分母）的整数部
        # 分转换为字符串，并添加到小数部分的字符串中
        remainder %= b

    print(f"小数点后第 {n} 位的数字是：{decimal[-1]}")
else:
    print("您输入的数字有误，请重启程序按照要求输入正确数字")

```

### 任务 5: Palindrome.py

```

N=int(input("请输入一个整数 N，以计算得到 1~N 之间满足条件的回文数："))

for num in range(1, N + 1):
    if int(num ** 0.5) ** 2 == num: # 判断是否为平方数
        # 判断是否为回文数
        num_oct = oct(num)[2:] # 将数字转换为八进制
        if num_oct == num_oct[::-1]: # 判断八进制是否为回文数
            print(num, end=' ')

```

## 五、 实验结果

（运行结果，截图）

## 任务 1

```
>>>
=== RESTART: D:\MineP\Program\Python.py\Modern_Program\Task3\Grade_grading.py ===
请输入百分制分数: 10
该分数分档为 D
>>>
=== RESTART: D:\MineP\Program\Python.py\Modern_Program\Task3\Grade_grading.py ===
请输入百分制分数: 59.9
该分数分档为 D
>>>
=== RESTART: D:\MineP\Program\Python.py\Modern_Program\Task3\Grade_grading.py ===
请输入百分制分数: 60
该分数分档为 C
>>>
=== RESTART: D:\MineP\Program\Python.py\Modern_Program\Task3\Grade_grading.py ===
请输入百分制分数: 74
该分数分档为 C
>>>
=== RESTART: D:\MineP\Program\Python.py\Modern_Program\Task3\Grade_grading.py ===
请输入百分制分数: 74.00001
该分数分档为 D
>>>
=== RESTART: D:\MineP\Program\Python.py\Modern_Program\Task3\Grade_grading.py ===
请输入百分制分数: 99.9
该分数分档为 A
```

## 任务 2

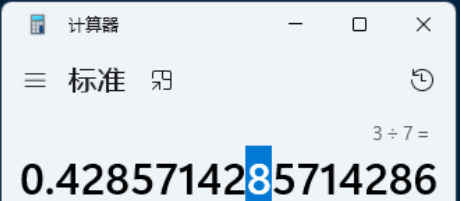
```
>>>
=== RESTART: D:\MineP\Program\Python.py\Modern_Program\Task3\Prime_number.py ===
请输入一个正整数 n: 5
2
3
5
7
11
>>>
=== RESTART: D:\MineP\Program\Python.py\Modern_Program\Task3\Prime_number.py ===
请输入一个正整数 n: 10
2
3
5
7
11
13
17
19
23
29
>>>
=== RESTART: D:\MineP\Program\Python.py\Modern_Program\Task3\Prime_number.py ===
请输入一个正整数 n: 20
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
53
```

### 任务 3

```
>>>
===== RESTART: D:\MineP\Program\Python.py\Modern_Progra
请输入一个正整数 n: 5
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
>>>
===== RESTART: D:\MineP\Program\Python.py\Modern_Progra
请输入一个正整数 n: 10
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
 1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1
```

### 任务 4

```
>>>
===== RESTART: D:\MineP\Program\Python.py\Modern_Program\Task3\Decimal.py =====
请输入分子 a (0 < a < b < 100) : 2
请输入分母 b (0 < a < b < 100) : 3
请输入要求的小数点后第 n 位 (1 <= n <= 10000) : 9887
小数点后第 9887 位的数字是: 6
>>>
===== RESTART: D:\MineP\Program\Python.py\Modern_Program\Task3\Decimal.py =====
请输入分子 a (0 < a < b < 100) : 3
请输入分母 b (0 < a < b < 100) : 7
请输入要求的小数点后第 n 位 (1 <= n <= 10000) : 9
小数点后第 9 位的数字是: 8
>>>
```



计算器

标准

3 ÷ 7 =

0.4285714285714286

### 任务 5



```

>>>
==== RESTART: D:\MineP\Program\Python.py\Modern_Program\Task3\Palindrome
请输入一个整数N，以计算得到1~N之间满足条件的回文数：100
1 4 9 36 81
>>>
==== RESTART: D:\MineP\Program\Python.py\Modern_Program\Task3\Palindrome
请输入一个整数N，以计算得到1~N之间满足条件的回文数：10000
1 4 9 36 81 121 729 4225 5329 6241 6561 6889
>>>
==== RESTART: D:\MineP\Program\Python.py\Modern_Program\Task3\Palindrome
请输入一个整数N，以计算得到1~N之间满足条件的回文数：10000000
1 4 9 36 81 121 729 4225 5329 6241 6561 6889 38025 47961 56169 133956 26316
5 485809 1196836 2368521 3080025 3515625 8468100

```

## 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.任务 1 复习理解了 if-elif-else 语句的使用。
- 2.任务 2 时熟悉了 for 循环、while 循环的使用，增强了对 if-else 语句的理解。
- 3.任务 3 学习了使用字符串处理数据、len()函数和格式化输出(左右对齐等)。
- 4.任务 4 学习了字符串的索引、字符串的格式化输出，理解长除法和取余的使用。
- 5.任务 5 学习了将十进制数转换为八进制数的函数 oct()、学习了索引切片。
- 6.增强了对 for 循环、while 循环、if-elif-else 语句的理解和使用，同时学习了一些函数和操作。

深圳大学学生实验报告用纸

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。