上海建桥学院

**本科实验报告**

|  |  |
| --- | --- |
| 课程名称： | 软件测试技术 |
| 学 号： | 2311806 |
| 姓 名： | 宋卓立 |
| 专 业： | 软件工程 |
| 班 级： | 软工B21-4 |
| 指导教师： | 龙永辉 |

**课内实验目录及成绩**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 实验名称 | 页码 | 成绩 |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |

信息技术学院

2024年 4月 6日

**上海建桥学院实验报告**

课程名称： 软件测试技术 实验类型： python练习

实验项目名称：

实验地点： 实验日期： 2024 年 4 月 6 日

1. 实验目的

循环和字符串的练习

1. 实验过程及结果

# 创建一个空列表score

score = []

# 1) 在score列表中依次追加10个数值

score.extend([68, 87, 92, 108, 76, 88, 54, 89, 76, 61])

# 2) 输出score中第三个元素的数值 和 第1~6个元素的数值

print("第三个元素的数值:", score[2])

print("第1~6个元素的数值:", score[:6])

# 3) 在score第3个元素前添加数值59

score.insert(2, 59)

# 4) 查询有多少个同学的成绩是76

count\_76 = score.count(76)

print("成绩是76的同学数:", count\_76)

# 5) 查询score中成绩是满分的同学的学号(假设第一个同学的学号是1980100，而同学的学号是连续的)

perfect\_score\_index = score.index(108)

first\_student\_number = 1980100

perfect\_score\_students = [first\_student\_number + perfect\_score\_index]

print("成绩是满分的同学的学号:", perfect\_score\_students)

# 6) 对score进行排序，输出最高分和最低分

sorted\_score = sorted(score)

highest\_score = sorted\_score[-1]

lowest\_score = sorted\_score[0]

print("最高分:", highest\_score)

print("最低分:", lowest\_score)

# 7) 颠倒score中元素的顺序后打印输出，但score本身不变

reversed\_score = score[::-1]

print("颠倒后的score:", reversed\_score)

# 8) 在score中追加数据88，然后删除score中的第一个值为88的元素

score.append(88)

score.remove(88)

# 9) 创建两个列表score1和score2

score1 = [8, 61]

score2 = [71, 95, 82]

# 合并这两个列表，并输出全部元素

combined\_score = score1 + score2

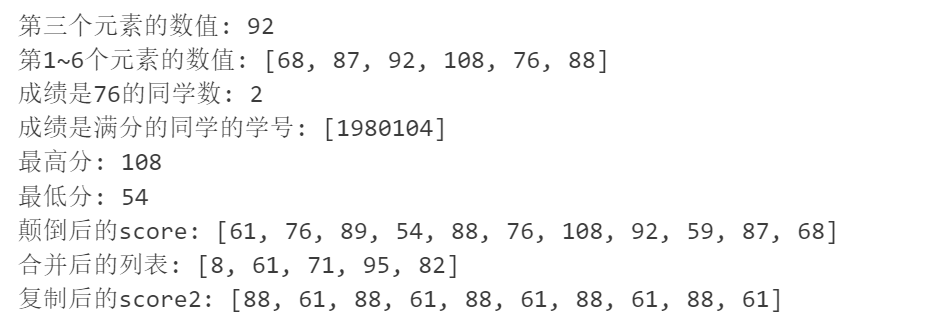
print("合并后的列表:", combined\_score)

# 10) 创建列表score1，将score1中的元素复制5遍保存到score2中

score1 = [88, 61]

score2 = score1 \* 5

print("复制后的score2:", score2)



观察到输出的矩阵大小为 n x n，并且从左上角开始，按照顺时针方向填充数字，每次填充完一圈后，填充的方向会发生改变。所以一开始要需要创建一个 n x n 的二维数组来存储填充的数字，然后按照顺时针方向依次填充数字，同时记录填充的方向，最后当一圈填充完毕后，需要改变填充方向，继续填充下一圈的数字，直到填充完整个矩阵。

def generate\_pattern(n):

    # 创建一个 n x n 的二维数组，初始化为0

    matrix = [[0] \* n for \_ in range(n)]

    num = 1  # 起始数字

    row, col = 0, 0  # 起始填充位置

    direction = 1  # 初始填充方向

    # 填充数字

    for i in range(1, n \* n + 1):

        matrix[row][col] = i  # 填充数字

        if direction == 1:  # 向右填充

            if col + 1 < n and matrix[row][col + 1] == 0:

                col += 1

            else:  # 改变方向

                direction = 2

                row += 1

        elif direction == 2:  # 向下填充

            if row + 1 < n and matrix[row + 1][col] == 0:

                row += 1

            else:  # 改变方向

                direction = 3

                col -= 1

        elif direction == 3:  # 向左填充

            if col - 1 >= 0 and matrix[row][col - 1] == 0:

                col -= 1

            else:  # 改变方向

                direction = 4

                row -= 1

        elif direction == 4:  # 向上填充

            if row - 1 >= 0 and matrix[row - 1][col] == 0:

                row -= 1

            else:  # 改变方向

                direction = 1

                col += 1

    # 输出矩阵

    for row in matrix:

        print(" ".join([str(num).zfill(2) for num in row]))

# 测试

generate\_pattern(4)  # 输出 4 的情况

generate\_pattern(5)  # 输出 5 的情况

