中国人民公安大学
实验报告

课程名称 C 语言程序设计

实验项目 实验二 数组程序设计

实验类型 设计性实验

姓　　名 胡文强

学　　号 202121710039

年　　级 2021 级

专　　业 数据警务技术

指导教师 邵翀

成　　绩

实验日期 2021 年 5 月 25 日

教 务 处 制

实验报告撰写说明

1.实验报告由参加实验的学生按要求独自完成。

2.课程名称、实验项目和实验类型要与实验教学大纲一致。

3.实验报告按验证性实验、设计性实验和综合性实验报告的不同要求来编写。其中，“实验设计方案（或实验综合知识）”项目验证性实验不用填写。

4.实验数据、曲线、图表和照片等要符合实验指导书的要求，分别插入实验报告的相应部分，内容比较多的也可以作为“附件”附在实验报告的后面。

一、实验内容

1. 编写程序，用筛选法求解200以内的所有素数并带格式输出。
2. 编写程序，输出一个杨辉三角形。
3. 编写程序，对餐饮质量评价进行统计输出。
4. 编写程序，比较两个字符串的大小。

二、实验目的和要求

1. 掌握一维数组、二维数组的定义和引用方法。
2. 掌握数据序列的排序方法。
3. 掌握字符串存储和处理方法。
4. 掌握字符处理函数的功能和应用方法。
5. 掌握在程序调试时查看数组内容的方法。

三、实验设备（软、硬件）

硬件设备：PC机
软件环境：Vim，GCC，GDB

四、实验设计方案（或实验综合知识）

（无）

五、实验原理

（同目的要求 略）

六、实验方法及步骤

分别使用 Vim 编辑器编辑源代码文件，后使用 gcc SOURCE_CODE -o TARGET_EXECUTABLE 命令编译源代码文件，并使用 ./TARGET_EXECUTABLE 命令运行生成的可执行文件。
程序运行结果未达预期时，使用 GDB 进行调试。在编译命令中加入 -g 参数表明生成供调试的可执行文件，后使用 gdb TARGER_EXECUTABLE 命令来调试该程序。

七、实验数据记录与处理

```
/**
* @File_Name: 1.c
* @Author: Wenqiang Hu
* @Mail: huwenqiang.hwq@protonmail.com
* @Created_Time: 2022 年 05 月 25 日 星期三 08 时 40 分 57 秒
* @Description: 输出 200 以内的素数
**/
```

```c
#include <stdio.h>

int main()
{
    int i = 0;
    int j = 0;
    int count = 1;
    int result[50]={0};
    result[0] = 2;

    for (i = 3; i <= 200; i++) // 2 是偶数，但是 2 却是素数
    {
        if (i % 2 == 0) //先进行判断，如果是偶数，直接跳过
            continue;
        else
        {
            for (j = 2; j < i; j++)
            {
                if (i % j == 0)
                    break;
            }
            if (i == j)
            {
                result[count] = i;
                count = count + 1;
            }
        }
    }

    count = 1;
    /*
    for (i = 0; i < 50; i++)
    {
        if (result[i] != 0)
        {
            if (count < 10)
            {
                printf("0%d: %d\n", count, result[i]);
                count = count + 1;
            }
            else
            {
                printf("%d: %d\n", count, result[i]);
                count = count + 1;
            }
        }
    }
    */
    for (i = 0; i < 50; i++)
    {
        if (result[i] != 0)
        {
            if (i % 7 == 0)
                printf("\n");
            printf("%-4d ", result[i]);
        }
    }
    return 0;
```

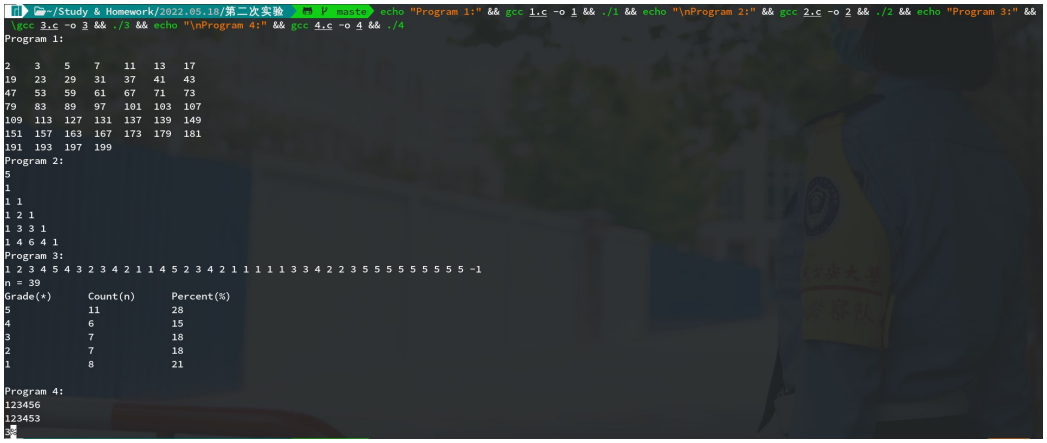
```
}



/**
* @File_Name: 2_1.c
* @Author: Wenqiang Hu
* @Mail: huwenqiang.hwq@protonmail.com
* @Created_Time: Wed May 25 09:15:34 2022
* @Description:
**/

#include <stdio.h>

int main()
{
    int i, j, k;
    int n=0;
    int a[50][50] = {0};

    scanf("%d",&n);

    for (i = 1; i <= n; i++)
        a[i][1] = a[i][i] = 1;

    for (i = 3; i <= n; i++)
    {
        for (j = 2; j <= i-1; j++)
            a[i][j] = a[i-1][j-1] + a[i-1][j]; /* 除两边的数外都等于上两
顶数之和 */
    }

    for (i = 1; i <= n; i++)
    {
        for (k = 1; k <= n-i; k++)
        {
            printf("   "); /* 使输出居中 */
        }
        for (j = 1; j <= i; j++)  /* j <= i 是为了不输出其它的数 */
        {
            printf("%6d", a[i][j]);
        }

        printf("\n");  /* 当一行输出完以后换行继续下一行的输出 */
    }
    printf("\n");
}



/**
* @File_Name: 3.c
* @Author: Wenqiang Hu
* @Mail: huwenqiang.hwq@protonmail.com
* @Created_Time: Wed May 25 10:01:55 2022
* @Description: 餐饮评价等级统计
**/
```

```c
#include <stdio.h>

int main()
{
    int valuation[1000] = {0};
    int count = 0;
    int input;
    int grade_1 = 0, \
        grade_2 = 0, \
        grade_3 = 0, \
        grade_4 = 0, \
        grade_5 = 0;

    while(1) {
        scanf ("%d", &input);
        if (input != -1) // 由于是理想输入，故不对输入内容的合法性进行校验
        {
            valuation[count] = input;
            count = count + 1;
        }
        else
        {
            break;
        }
    }

    for (int i = 0; i < count; i++)
    {
        switch (valuation[i])
        {
            case 1:
                grade_1 ++; break;
            case 2:
                grade_2 ++; break;
            case 3:
                grade_3 ++; break;
            case 4:
                grade_4 ++; break;
            case 5:
                grade_5 ++; break;
        }
    }

    printf("n = %d\n", count);
    printf("Grade(*)      Count(n)      Percent(%)\n");
    printf("5             %-5d         %-2.0f\n", grade_5,
(float)((float)grade_5 * 100 / (float)count));
    printf("4             %-5d         %-2.0f\n", grade_4,
(float)((float)grade_4 * 100 / (float)count));
    printf("3             %-5d         %-2.0f\n", grade_3,
(float)((float)grade_3 * 100 / (float)count));
    printf("2             %-5d         %-2.0f\n", grade_2,
(float)((float)grade_2 * 100 / (float)count));
    printf("1             %-5d         %-2.0f\n", grade_1,
(float)((float)grade_1 * 100 / (float)count));
```

```c
    return 0;
}

/**
* @File_Name: 4.c
* @Author: Wenqiang Hu
* @Mail: huwenqiang.hwq@protonmail.com
* @Created_Time: 2022 年 05 月 25 日 星期三 10 时 56 分 13 秒
* @Description: String Compare
**/

#include <stdio.h>
#include <string.h>

int strcmp_self(char *str_1, char *str_2)
{
    while ((*str_1 != '\0') && (*str_1 == *str_2))
    {
        str_1++;
        str_2++;
    }
    int result = *str_1 - *str_2;
    return result;
}

int main()
{
    char str_1[100], str_2[100];

    char * string_1 = str_1;
    char * string_2 = str_2;

    /* 直接使用 gets() 会引入缓冲区溢出的风险，所以现在换成了 fgets()*/
    fgets(string_1, 100, stdin);
    fgets(string_2, 100, stdin);

    printf("%d", strcmp_self(str_1, str_2));
    return 0;
}
```

八、实验结果及分析

所有源代码程序一次性编译运行后的结果如下图所示：



本次实验主要在 Manjaro Linux 环境下进行，使用 Vim 8.2 作为主要编辑器，使用 GCC 11.2.0 作为编译器，同时使用 GDB 11.2 作为程序调试器，使用 zsh 作为 SHELL 解释器。

本实验共涉及格式化输出、数组运用、循环下标控制、指针配合等前置知识，同时部分程序可能涉及相应的调试技巧。本次实验中的程序编写难度相较第一次明显增大，过程较为繁琐，极易在细节处犯错。经过多次核对、调试后，程序均得到预想输出，实验基本成功。

九、实验总结

本次实验总体较为成功，但通过同学们的讨论可以发现：对于 printf 函数的格式化输出掌握不足，尤其是涉及左右对齐和位数控制时，导致程序输出不达预期；部分同学使用 Visual Studio 2022 IDE，该 IDE 使用 cl 作为默认编译器，且默认创建 C++ 源代码文件，在该情况下，编译器会对 C 语言标准库中的部分函数报警（如会认为 scanf 不安全，需替换为 scanf_s 等），同时可能涉及一些数值计算的问题；另外，同学们对于调试的掌握不足，在程序出现错误时不会主动进行调试。

在该实验中，我更加深入地学习了 C 语言的经典例题，提高了编码能力与速度，且接触了更多因编译器不同导致的特殊 bug。

十、参考资料

《The C Programming Language》
《深入理解计算机系统》
《啊哈算法》
《C Primer Plus》
《C 语言程序设计》

十一、指导教师评语及成绩