

程序分组设计训练 实验 3

实验报告

学期：2022-2023 第二学期

学院：计算机与信息技术学院

姓名：张鲔沅

学号：22281052

班级：计算机 2202 班

编制日期：2023 年 4 月 18 日

目 录

实验 实现文件自动生成

1 文件生成——结构体与动态内存申请	1
1.1 结果展示	1
1.1.1 软件调试结果	1
1.1.2 软件调试结果	2
1.2 相关问题回答	6
1.2.1 问题 a	6
1.2.2 问题 b	10
1.2.3 问题 c	20
1.2.4 问题 d	22
1.2.5 问题 e	23
1.2.6 问题 f	24
1.2.7 问题 g	25
1.2.8 问题 h	26

实验总结

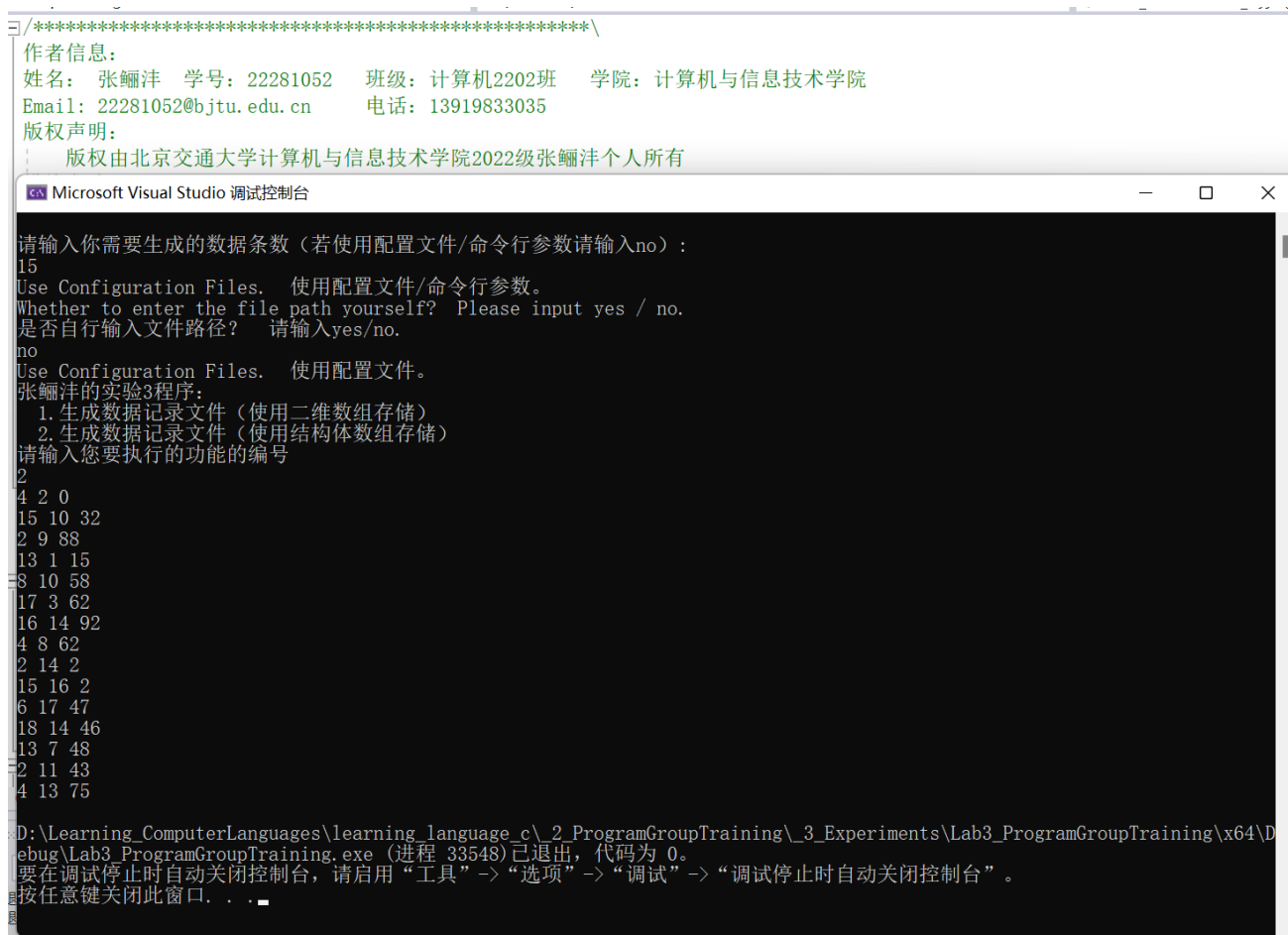
2 遇到的问题及解决办法	27
2.1 程序与调试相关问题	27
2.1.1 指针不清晰——LNK1120、LNK2019 报错	27
2.1.2 cmd 窗口与工作目录	29
3 实验的收获与心得	30
参考文献	30
相关附件:	30

实验 文件生成——结构体与动态内存申请

1 文件生成——结构体与动态内存申请

1.1 结果展示

1.1.1 软件调试结果^①



```

/*****\
作者信息：
姓名： 张颢沅  学号： 22281052  班级： 计算机2202班  学院： 计算机与信息技术学院
Email: 22281052@bjtu.edu.cn  电话： 13919833035
版权声明：
    版权由北京交通大学计算机与信息技术学院2022级张颢沅个人所有

Microsoft Visual Studio 调试控制台

请输入你需要生成的数据条数（若使用配置文件/命令行参数请输入no）：
15
Use Configuration Files. 使用配置文件/命令行参数。
Whether to enter the file path yourself? Please input yes / no.
是否自行输入文件路径？ 请输入yes/no.
no
Use Configuration Files. 使用配置文件。
张颢沅的实验3程序：
    1. 生成数据记录文件（使用二维数组存储）
    2. 生成数据记录文件（使用结构体数组存储）
请输入您要执行的功能的编号
2
4 2 0
15 10 32
2 9 88
13 1 15
8 10 58
17 3 62
16 14 92
4 8 62
2 14 2
15 16 2
6 17 47
18 14 46
13 7 48
2 11 43
4 13 75

D:\Learning_ComputerLanguages\learning_language_c\2_ProgramGroupTraining\3_Experiments\Lab3_ProgramGroupTraining\x64\Debug\Lab3_ProgramGroupTraining.exe (进程 33548) 已退出，代码为 0。
要在调试停止时自动关闭控制台，请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。
按任意键关闭此窗口。 . . .

```

图 1-1-1 Visual Studio 2022 运行结果

^① 调试软件为 Visual Studio 2022。

1.1.2 软件调试结果

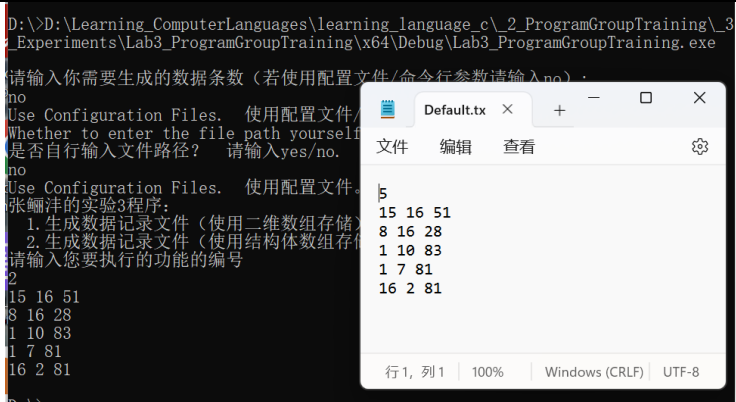
结果展示数据源如下表：

表 1 用户输入与实际文件生成情况举例

配置文件初始内容		用户输入		程序实际执行后记录文件生成情况	
文件的存储目录	文件名	命令行参数	运行后输入	文件的存储目录	生成文件名
D:/Lab3/Output	Default.txt	Lab3.exe	<选择默认>	D:/Lab3/Output	Default.txt
D:/Lab3/Output	Default.txt	Lab3.exe	111.txt	D:/Lab3/Output	111.txt
D:/Lab3/Output	Default.txt	Lab3.exe	D:/aaa/222.txt	D:/aaa	222.txt
D:/Lab3/Output	Default.txt	Lab3.exe	../333.txt	exe 当前目录的上一级	333.txt
D:/Lab3/Output	Default.txt	Lab3.exe 100	<选择默认>	D:/Lab3/Output	Default.txt
D:/Lab3/Output	Default.txt	Lab3.exe 100	555.txt	D:/Lab3/Output	555.txt
D:/Lab3/Output	Default.txt	Lab3.exe 100	E:/222/abc.txt	E:/222	abc.txt
D:/Lab3/Output	Default.txt	Lab3.exe 100	/sub/666.txt	exe 当前盘根目录下的 sub 子目录	666.txt
D:/Lab3/Output	Default.txt	Lab3.exe out.txt		D:/Lab3/Output	out.txt
D:/Lab3/Output	Default.txt	Lab3.exe F:/out.txt		F:/	out.txt
D:/Lab3/Output	Default.txt	Lab3.exe ../out.txt		exe 当前目录的上一级	out.txt
D:/Lab3/Output	Default.txt	Lab3.exe 100 ok.txt		D:/Lab3/Output	ok.txt
D:/Lab3/Output	Default.txt	Lab3.exe 100 D:/a/ok.txt		D:/a	ok.txt
D:/Lab3/Output	Default.txt	Lab3.exe 100 b/ok.txt		exe 当前目录下的 b 子目录	ok.txt

结果展示如下：

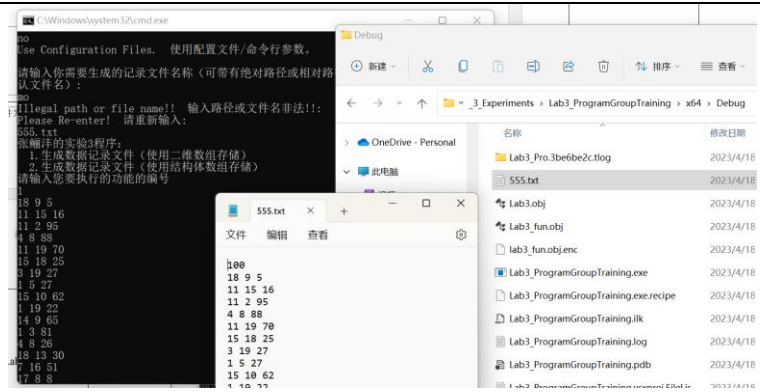
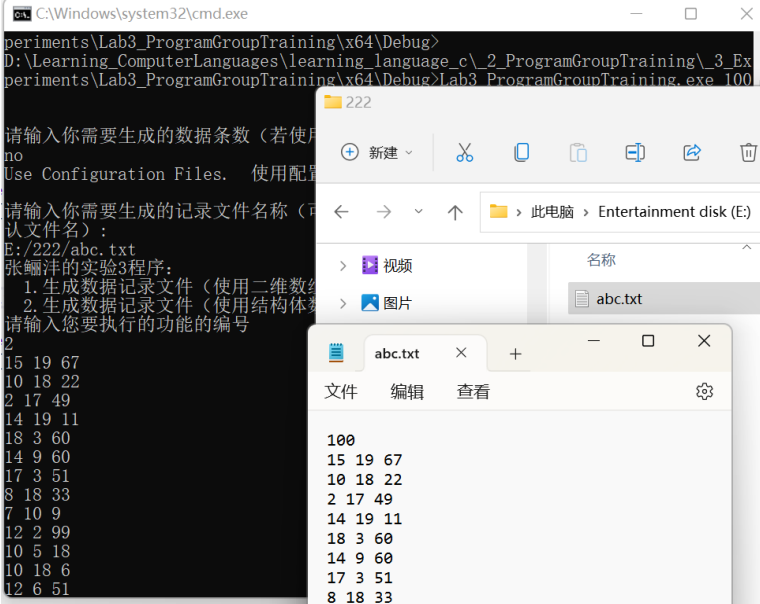
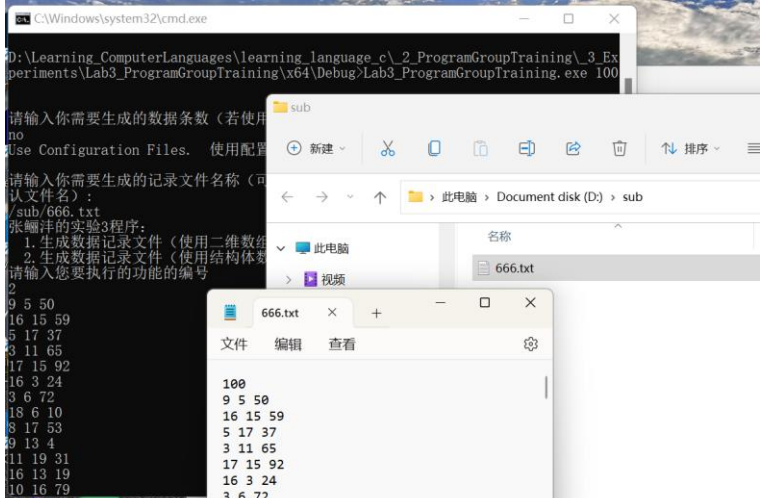
表 2 结果截图情况表

命令行参数	运行后输入	结果
Lab3.exe	<选择默认>	

续表 2

Lab3.exe	111.txt	
Lab3.exe	D:/aaa/222.txt	
Lab3.exe	../333.txt	
Lab3.exe 100	<选择默认>	

续表 2

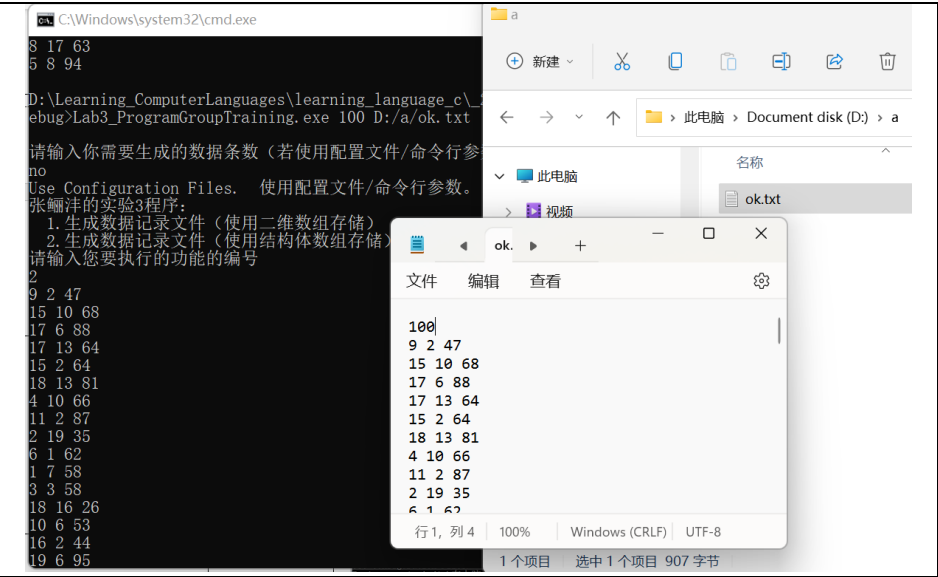
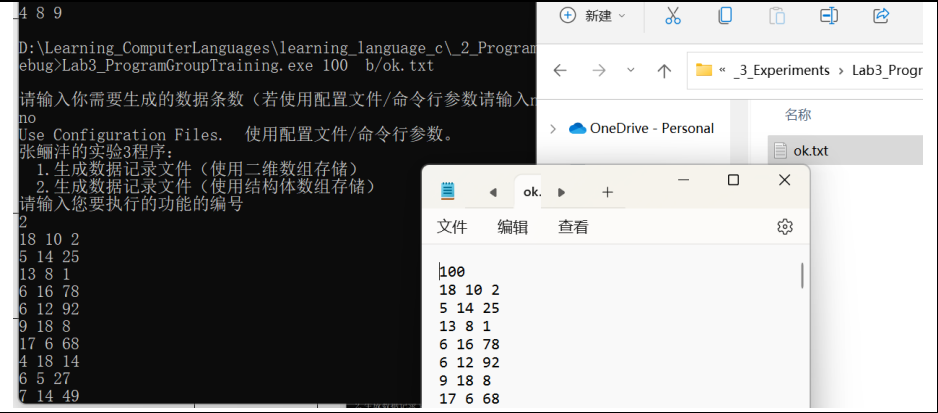
Lab3.exe 100	555.txt	 <p>The screenshot shows a Windows command prompt window with the following text:</p> <pre> C:\Windows\system32\cmd.exe Use Configuration Files. 使用配置文件/命令行参数。 请输入你需要生成的记录文件名称 (可带有绝对路径或相对路 默认文件名): Illegal path or file name!! 输入路径或文件名非法!! Please Re-enter! 请重新输入: 555.txt 张颉泮的实验3程序: 1. 生成数据记录文件 (使用二维数组存储) 2. 生成数据记录文件 (使用结构体数组存储) 请输入您要执行的功能的编号 2 18 9 5 11 15 16 11 2 95 14 8 88 11 19 70 15 18 25 11 5 27 15 10 62 11 19 22 14 9 65 13 3 51 14 8 26 18 13 30 17 16 51 17 8 8 </pre> <p>A file explorer window shows the file 555.txt in the directory C:_3_Experiments\Lab3_ProgramGroupTraining\Debug.</p>
Lab3.exe 100	E:/222/abc.txt	 <p>The screenshot shows a Windows command prompt window with the following text:</p> <pre> C:\Windows\system32\cmd.exe periments\Lab3_ProgramGroupTraining\x64\Debug> D:\Learning_ComputerLanguages\learning_language_c\2_ProgramGroupTraining\3_Ex periments\Lab3_ProgramGroupTraining\x64\Debug>Lab3_ProgramGroupTraining.exe 100 请输入你需要生成的数据条数 (若使用 no Use Configuration Files. 使用配置 请输入你需要生成的记录文件名称 (可 默认文件名): E:/222/abc.txt 张颉泮的实验3程序: 1. 生成数据记录文件 (使用二维数组存储) 2. 生成数据记录文件 (使用结构体数组存储) 请输入您要执行的功能的编号 2 15 19 67 10 18 22 2 17 49 14 19 11 18 3 60 14 9 60 17 3 51 8 18 33 7 10 9 12 2 99 10 5 18 10 18 6 12 6 51 </pre> <p>A file explorer window shows the file abc.txt in the directory E:\222.</p>
Lab3.exe 100	/sub/666.txt	 <p>The screenshot shows a Windows command prompt window with the following text:</p> <pre> C:\Windows\system32\cmd.exe D:\Learning_ComputerLanguages\learning_language_c\2_ProgramGroupTraining\3_Ex periments\Lab3_ProgramGroupTraining\x64\Debug>Lab3_ProgramGroupTraining.exe 100 请输入你需要生成的数据条数 (若使用 no Use Configuration Files. 使用配置 请输入你需要生成的记录文件名称 (可 默认文件名): /sub/666.txt 张颉泮的实验3程序: 1. 生成数据记录文件 (使用二维数组存储) 2. 生成数据记录文件 (使用结构体数组存储) 请输入您要执行的功能的编号 2 9 5 50 16 15 59 5 17 37 3 11 65 17 15 92 16 3 24 3 6 72 18 6 10 18 17 53 19 13 4 11 19 31 16 13 19 10 16 79 </pre> <p>A file explorer window shows the file 666.txt in the directory D:\sub.</p>

续表 2

Lab3.exe out.txt		 <p>请输入你需要生成的数据条数 (若使用配置文件/命令行参数请输入no): no Use Configuration Files. 使用配置文件/命令行参数。 1. 生成数据记录文件 (使用二维数组存储) 2. 生成数据记录文件 (使用结构体数组存储) 请输入您要执行的功能的编号 14 9 32 6 16 62 4 13 36 15 9 68 1 10 72</p> <p>out.txt</p> <p>文件 编辑 查看</p> <p>Lab3_run.obj out.txt vc143.idb vc143.pdb</p>
Lab3.exe F:/out.txt ^①		 <p>请输入你需要生成的数据条数 (若使用配置文件/命令行参数请输入no): no Use Configuration Files. 使用配置文件/命令行参数。 F: is no exist! disk在电脑中不存在 Illegal Command Line Arguments! 命令行参数非法! Program ending! 程序结束!</p> <p>D:\Learning_ComputerLanguages\learning_language_c\2_ProgramGroupTraining\3_Experiments\Lab3_ProgramGroupTraining\x64\Debug></p>
Lab3.exe F:/out.txt		 <p>请输入你需要生成的数据条数 (若使用配置文件/命令行参数请输入no): no Use Configuration Files. 使用配置文件/命令行参数。 张颀洋的实验3程序: 1. 生成数据记录文件 (使用二维数组存储) 2. 生成数据记录文件 (使用结构体数组存储) 请输入您要执行的功能的编号 12 11 84 7 17 50 13 1 18 17 3 71 9 18 67</p> <p>out.txt</p> <p>文件 编辑 查看</p> <p>行 1, 列 1 100% Windows (CRLF) UTF-8</p>
Lab3.exe ../out.txt		 <p>Microsoft Windows [版本 10.0.22000.1574] (c) Microsoft Corporation. 保留所有权利。 C:\Users\20957>cd D:\Learning_ComputerLanguages\learning_language_c\2_ProgramGroupTraining\3_Experiments\Lab3_ProgramGroupTraining\x64\Debug D:\Learning_ComputerLanguages\learning_language_c\2_ProgramGroupTraining\3_Experiments\Lab3_ProgramGroupTraining\x64\Debug>Lab3_ProgramGroupTraining.exe ../out.txt 请输入你需要生成的数据条数 (若使用配置文件/命令行参数请输入no): no Use Configuration Files. 使用配置文件/命令行参数。 张颀洋的实验3程序: 1. 生成数据记录文件 (使用二维数组存储) 2. 生成数据记录文件 (使用结构体数组存储) 请输入您要执行的功能的编号 15 3 64 315 17 44 15 17 65 5 2 73 6 17 41</p> <p>out.txt</p> <p>文件 编辑 查看</p>
Lab3.exe 100 ok.txt		 <p>请输入你需要生成的数据条数 (若使用配置文件/命令行参数请输入no): no Use Configuration Files. 使用配置文件/命令行参数。 张颀洋的实验3程序: 1. 生成数据记录文件 (使用二维数组存储) 2. 生成数据记录文件 (使用结构体数组存储) 请输入您要执行的功能的编号 100 3 11 26 10 8 62 12 16 32 16 13 94 15 6 96 3 1 87 1 14 89</p> <p>ok.txt</p> <p>文件 编辑 查看</p>

① 测试电脑上没有 F 盘, 因此报错。

续 表 2

Lab3.exe 100 D:/a/ok.txt	
Lab3.exe 100 b/ok.txt	

1.2 相关问题回答

1.2.1 问题 a

实验 3 采用新建工程方式构建，对于实验 2 中的已有程序，将可以复用的代码拷贝复制到实验 3 进行使用。请在实验报告中详细说明实验 3 中复用了实验 2 中的哪些函数，复用到实验 3 中的函数是完全复用还是拷贝后进行了必要的改动，请详细说明一下为了完成实验 3 复用的函数做了哪些修改。

答：

实验 3 中复用了以下函数：

1. int clearstdin()——完全复用

作用是对 scanf 输入进行缓存清除。

```

/*
 *函数名称: clearstdin
 *函数功能: 对scanf输入进行缓存清除
 *输入参数: 无
 *返回值: int(本程序无实际意义)
 *版本信息: create by Lifeng Zhang, 2022-03-17
 */
int clearstdin()
{
    int c;
    while ((c = getchar()) != '\n' && c != EOF)
        ;
    return 0;
}
```

图 1-2-1-1 Lab3 clearstdin 函数

2. int check_num(char* str)——部分复用

作用是对输入数据数据是否合法（为纯数字）进行检查。

增加了语句：**data->number = atoi(str);** 数字判断完成后直接进行存储。

```

/*
 *函数名称: check_num
 *函数功能: 对输入数据数据是否合法（为纯数字）进行检查
 *输入参数: 无
 *返回值: int, 如果检查后合法, 返回值为1, 否则为0
 *版本信息: create by Lifeng Zhang, 2022-04-01
 *           repair by Lifeng Zhang, 2023-04-18
 */
int check_num(CONF* data, char* str)
{
    int len = strlen(str);
    int i;
    for (i = 0; i < len; i++)
    {
        if (str[i] < '0' || str[i] > '9')
            return 0;
    }
    data->number = atoi(str);
    return 1;
}
```

图 1-2-1-2 Lab3 check_num 函数

3. void create_1(FILE* fp, int* num, char* argv[], CONF* data)

——部分复用

复用 Lab2 中的 create_1 函数。增添了以下内容：

询问是否输入文件。如果输入 yes，进入 get_filename 函数；反之不进行。

```

/*
 *函数名称: create_1
 *函数功能: 用户未输入数据情况、创建文件
 *输入参数: FILE*fp:文件指针、int* num:argv[1]文件条数、char* argv[]:命令行参数值
 *返回值: void
 *版本信息: create by Lifeng Zhang, 2022-03-17
 */
void create_1(FILE* fp, int* num, char* argv[])
{
    int number;
    printf("Please input the date of number 请输入数据行的个数:");
    scanf("%d", &number);
    clearstdin(); // 清缓存, 防止爆栈
    if ((fp = fopen("output.txt", "w")) == NULL)
        // 创建一个名称是 "output" 的txt文档
        printf("Error on open output.txt 未成功创建output.txt文件\n"); // 如果没有成功创建文档, 进行报错提示
    else
    {
        *num = number;
        data(fp, num, argv);
    }
}

```

图 1-2-1-3 Lab2 create_1 函数

```

/*
 *函数名称: create_1
 *函数功能: 用户未输入数据情况、创建文件
 *输入参数: FILE*fp:文件指针、int* num:argv[1]文件条数、char* argv[]:命令行参数值
 *返回值: void
 *版本信息: create by Lifeng Zhang, 2023-03-17
 */
void create_1(FILE* fp, int* num, char* argv[], CONF* data)
{
    get_filenum(data); // 获取文件数目

    char crea[4];
    printf("Whether to enter the file path yourself? Please input yes / no.\n");
    printf("是否自行输入文件路径? 请输入yes/no.\n");
    scanf("%s", crea);
    clearstdin();

    if (strcmp(crea, "no") == 0) // 不自行输入
        printf("Use Configuration Files. 使用配置文件.\n");
    else
        get_filename(data); // 获取文件名

    // 创建文档
    if ((fp = fopen(user_filename, "w")) == NULL)
        printf("Error on open filelet 未成功创建文件\n"); // 如果没有成功创建文档, 进行报错提示
    else
    {
        *num = data->number;
        writenum(fp, data, user_filename);
    }
}

```

图 1-2-1-4 Lab3 create_1 函数

4. void create_3(FILE* fp, int* num, char* argv[], CONF* data)

——部分复用

复用 Lab2 中的 create_2 函数。增添了以下内容：

- 1.判断命令行参数先是数字还是文件路径，进行相关操作。
- 2.联合其他函数，进行命令行参数合法性检查。

```

/*
*函数名称: create_2
*函数功能: 用户已输入数据情况、创建文件
*输入参数: FILE*fp:文件指针、int* num:argv[1]文件条数、char* argv[]:命令行参数值
*返回值: void
*版本信息: create by Lifeng Zhang, 2022-03-17
*/
void create_2(FILE* fp, int* num, char* argv[])
{
    if (check(argv[1]) == 0) // 用户输入第一个的非数字参数，数据有问题
        printf("Input error! 输入错误! \nProgram ending! 程序结束! \n");
    else // 用户输入正确
    {
        if ((fp = fopen(argv[2], "w")) == NULL)
            // 创建一个名称是输入的argv[2]的txt文档
            printf("Error on open put.txt 未成功创建output.txt文件\n"); // 如果没有成功创建文档，进行报错提示
        else
        {
            *num = atoi(argv[1]); // 将输入的char型数据转换为int型
            data(fp, num, argv);
        }
    }
}

```

图 1-2-1-7 Lab2 create_2 函数

```

void create_3(FILE* fp, int* num, char* argv[], CONF* data)
{
    char* num_file = NULL;
    char* name_file; // 初始化定义，以便后续判断
    get_filename(data); // 获取文件数目

    if (check_num(data, argv[1]) == 0) // 检查用户输入第一个参数
    {
        if (check_num(data, argv[2]) == 0) // 输入第一个参数是非数字，检查第二个数据
        {
            printf("The number of records parameter could not be found in the input parameter! 输入参数中找不到记录条数参数! \n");
            printf("Program ending! 程序结束! \n");
        }
        else
        {
            strcpy(user_filename, argv[1]);
            if (check_seper_file(argv[1], data) == 0)
            {
                printf("Illegal Command Line Arguments! 命令行参数非法! \n");
                printf("Program ending! 程序结束! \n");
            }
            else
                name_file = argv[2]; // 第二个参数为为文件名，进行储存
        }
    }
    else // 用户输入第一个参数是数字

```

图 1-2-1-8 Lab3 create_3 函数

1.2.2 问题 b

请在实验报告中论述实验 3 相比实验 2 多了哪些函数（实验 2 复用过来做了修改的函数不算），这些新设计的函数分别用于支撑实验 3 中的那些功能扩展或新增功能。

答：

1. void ini_read(CONF* data)

此函数支撑实验 3 中的进行文件配置功能。文件在进行 create 系列函数^①前进行，先期配置，后续可以修改相关数据。

如果无法读取配置文件，系统自行进行配置。

```
/*
 *函数名称: ini_read
 *函数功能: 前期配饰文件
 *输入参数: CONF* data:结构体指针data
 *返回值: void
 *版本信息: create by Lifeng Zhang, 2023-03-17
 */
void ini_read(CONF* data)
{
    FILE* fp = fopen("D:/Learning_ComputerLanguages/set/conf.ini", "r"); // 将文件地址打开，读取内容
    if (fp)
    {
        // 将对应的内容读入结构体
        fscanf(fp, "%s%d%d%d%d",
            data->filesavepath,
            data->filename,
            &data->number,
            &data->maxvalue1,
            &data->minvalue1,
            &data->maxvalue2,
            &data->minvalue2,
            &data->recordcount1,
            &data->recordcount2);
        fclose(fp); // 关闭文件
        fflush(fp); // 清缓存，防止爆栈
    }
    else
    {
        printf("未能打开打开文件conf.ini, 系统自动配置相关文件\n");
        // 程序给出默认值进行配置
        strcpy(data->filesavepath, "output/");
        strcpy(data->filename, "Default.txt");
    }
}
```

图 1-2-2-1 Lab3 ini_read 函数

① create 系列函数指 create_1、create_2、create_3 函数

2. void create_2(FILE* fp, int* num, char* argv[], CONF* data)

此函数支撑实验 3 中，用户只输入一个命令行参数，对此进行处理。

先进行判断命令行参数是数字还是地址，之后对缺的另一项参数进行提醒，联合其他语句，让用户进行输入存储。

```
/*
*函数名称: create_2
*函数功能: 用户已输入数据情况、创建文件
*输入参数: FILE*fp:文件指针、int* num:文件条数、char* argv[]:命令行参数值、CONF data:结构体数组地址
*返回值: void
*版本信息: create by Lifeng Zhang, 2023-04-04
*          repair by Lifeng Zhang, 2023-04-18
*/
void create_2(FILE* fp, int* num, char* argv[], CONF* data)
{
    char* name_file = NULL; // 初始化定义，以便后续判断
    get_filenum(data); // 获取文件数目

    if (check_num(data, argv[1]) == 0) // 检查用户输入参数
    {
        // 输入参数是路径，检查数据
        strcpy(user_filename, argv[1]);
        if (check_seper_file(argv[1], data) == 0)
        {
            printf("Illegal Command Line Arguments! 命令行参数非法! \n");
            printf("Program ending! 程序结束! \n");
        }
        else
        {
            name_file = argv[1];
        }
    }
    else // 输入参数是数目，检查文件名
    {
        get_filename(data); // 获取文件名
        name_file = argv[1];
    }
    if (name_file == NULL)
```

图 1-2-2-2 Lab3 create_2 函数

3. void get_filenum(CONF* data)

此函数支撑实验 3 中，用户只缺少文件记录条数参数情况，提醒输入处理。

用户输入后，进入到 check_num 函数进行文件名检查，如果数字错误，进行报错，并且让用户重新输入。

如果用户使用自行输入的命令行参数或者使用配置文件，则输入 no，相当于跳过程序（输入的相关参数已经进行存储）。

```
/*
*函数名称: get_filenum
*函数功能: 通过交互界面获取文件记录条数
*输入参数: CONF data:结构体数组地址
*返回值: int, 如果检查后合法, 返回值为1, 否则为0
*版本信息: create by Lifeng Zhang, 2023-04-04
*          repair by Lifeng Zhang, 2023-04-18
*/
void get_filenum(CONF* data)
{
    char user_number[MAX_STR_LEN] = "no";
    printf("\n请输入你需要生成的数据条数（若使用配置文件/命令行参数请输入no）:\n");
    scanf("%s", user_number);
    clearstdin();

    // 判断用户是否输入合法
    if (strcmp(user_number, "no") != 0)
    {
        while (check_num(data, user_number) == 0)
        {
            printf("Illegal Number!! 输入数字非法!!:\n");
            printf("Please Re-enter! 请重新输入:\n");
            scanf("%s", user_number);
            clearstdin();
            if (strcmp(user_number, "no") == 0)
                break;
        }
        if (check_num(data, user_number) != 0)
            // 前期已经初始化, 写入了数据
            printf("Use Configuration Files. 使用配置文件/命令行参数.\n");
    }
    else
        // 前期已经初始化, 写入了数据
        printf("Use Configuration Files. 使用配置文件/命令行参数.\n");
}
```

图 1-2-2-3 Lab3 get_filenum 函数

4. void get_filename(CONF* data)

此函数支撑实验 3 中，用户只缺少文件储存路径参数情况，提醒输入处理。

用户输入后，进入到 check_seper_file 函数进行文件名检查，如果文件名错误，进行报错，并且让用户重新输入。

如果用户使用自行输入的命令行参数或者使用配置文件，则输入 no，相当于跳过程序（输入的相关参数已经进行存储）。

```
/*
 *函数名称: get_filename
 *函数功能: 通过交互界面获取文件储存路径
 *输入参数: char* str:文件路径与文件名 CONF data:结构体数组地址
 *返回值: int, 如果检查后合法, 返回值为1, 否则为0
 *版本信息: create by Lifeng Zhang, 2023-04-04
 *          repair by Lifeng Zhang, 2023-04-18
 */

void get_filename(CONF* data)
{
    printf("\n请输入你需要生成的记录文件名称（可带有绝对路径或相对路径，输入no表示使用配置默认文件名）:\n");
    scanf("%s", user_filename);
    clearstdin();

    // 判断用户是否输入合法
    if (strcmp(user_filename, "no") != 0)
    {
        while (check_seper_file(user_filename, data) == 0)
        {
            printf("Illegal path or file name!! 输入路径或文件名非法!!:\n");
            printf("Please Re-enter! 请重新输入:\n");
            scanf("%s", user_filename);
            clearstdin();
            if (strcmp(user_filename, "no") == 0)
                break;
        }
    }
    else
        printf("Use Configuration Files. 使用配置文件.\n");
}
```

图 1-2-2-4 Lab3 get_filename 函数

5. int check_seper_file(char* str, CONF* data)

此函数支撑实验 3 中， argv[1]或 argv[2]数据是否合法（为正确文件名）进行检查，判断路径为相对路径还是绝对路径，之后切割路径与文件名，进行储存。

先对整体进行检查，如果有“实验要求文件以.txt 结尾”“以“\”或“/”结尾”的情况，结束程序进行报错。之后判断是否是绝对路径。

```

/*
*函数名称: check_seper_file
*函数功能: 对argv[1]或argv[2]数据是否合法（为正确文件名）进行检查
*          判断路径为相对路径还是绝对路径，之后切割路径与文件名，进行储存
*输入参数: char* str:文件路径与文件名  CONF data:结构体数组地址
*返回值:   int, 如果检查后合法，返回值为1, 否则为0
*版本信息: create by Lifeng Zhang, 2023-04-01
*          repair by Lifeng Zhang, 2023-04-04
*/
int check_seper_file(char* str, CONF* data)
{
    int len = strlen(str);
    // _MAX_DRIVE、_MAX_DIR、_MAX_FNAME、_MAX_EXT 在配置中已有定义
    // _MAX_DRIVE==3, _MAX_DIR==_MAX_FNAME==_MAX_EXT==256
    char disk[_MAX_DRIVE];
    char dir[_MAX_DIR];
    char fname[_MAX_FNAME];
    char ext[_MAX_EXT];

    char way[MAX_STR_LEN];
    char file[MAX_STR_LEN];
    char filename_t[MAX_STR_LEN] = "";

    // 实验要求文件以.txt结尾
    if (str[len - 1] != 't' && str[len - 2] != 'x' && str[len - 3] != 't' && str[len - 4] != '.')
        return 0;

    if (*str == '\\') // 不能以“\”开头
        return 0;

    // 不能以“\”或“/”结尾，若以此结尾则认为文件名为空
    char* index1 = strrchr(str, '\\');
    char* index2 = strrchr(str, '/');
    if (index1 - str + 1 == strlen(str) || index2 - str + 1 == strlen(str))
        return 0;

    // 进行合并，以便后续处理
    char* index;
    if (index2 > index1)
        index = index2;
    else
        index = index1;

    // 判断输入路径为绝对路径还是相对路径
    char* p1 = strstr(str, ":\\");
    char* p2 = strstr(str, ":/");

```

图 1-2-2-5 Lab3 get_filename 函数第一部分

如果是，则运用_splitpath 函数^①进行分隔。之后判断盘符是否存在，文件路径是否合法。不合法结束程序进行报错。合法则将数据运用_makepath 函数^②，录入到结构体中。

如果不是，判断文件路径是否合法不合法结束程序进行报错。合法则将数据运录入到结构体中。

```

3 // 储存路径、文件名
// void _makepath( const char *path, char *drive, char *dir, char *fname, char *ext); #include <direct.h>
// 第一个是要储存的完整的文件名路径
// 四个参数分别代表四个需要从结合字符串，有驱动器盘符(drive)，中间的路径 (dir)，文件名(fname)，和后缀名 (ext)
_makepath(way, disk, dir, NULL, NULL);
_makepath(file, NULL, NULL, fname, ext);

strcpy(data->filesavepath, way);
strcpy(data->filename, file);

if (access(data->filesavepath, 0) == -1) //路径不存在，创建新路径
    mkdir(data->filesavepath);
}

else if (p1 == NULL || p2 == NULL) //仅仅是文件名
{
    if (strstr(str, "*") || strstr(str, "?") || strstr(str, "<") || strstr(str, ">") || strstr(str, "|"))
        return 0;

    strcpy(data->filename, str);
}

else //相对路径
{
    if (strstr(str, "*") || strstr(str, "?") || strstr(str, "<") || strstr(str, ">") || strstr(str, "|"))

```

图 1-2-2-6 Lab3 get_filename 函数第二部分

① void _splitpath(const char *path, char *drive, char *dir, char *fname, char *ext)。

第一个是待处理的完整的文件名路径，四个参数分别代表四个需要从原始文件路径中截取的字符串，有驱动器盘符(drive)，中间的路径 (dir)，文件名(fname)，和后缀名 (ext)

② void _makepath(const char *path, char *drive, char *dir, char *fname, char *ext)，头文件是#include <direct.h>。

第一个是要储存的完整的文件名路径，四个参数分别代表四个需要从结合字符串，有驱动器盘符(drive)，中间的路径 (dir)，文件名(fname)，和后缀名 (ext)

6. int writenum(FILE* fp, CONF* data, char* temp)

此函数支撑实验 3 中，菜单显示功能与数据写入文件功能，其中包括动态内存分配申请结构体，进行数据录入。

先让用户选择存储方式。之后进行随机数写入。

```
/*
 * 函数名称: int writenum
 * 函数功能: 判断需要的存储方式
 * 输入参数: CONF* data: 结构体指针data; char* temp:文件名
 * 返回值: int(返回0则结束此处程序)
 * 版本信息: create by Lifeng Zhang, 2023-04-17
 *           repair by Lifeng Zhang, 2023-04-18
 */
int writenum(FILE* fp, CONF* data, char* temp)//temp 文件名
{
    int way = 0;
    printf("张颢洋的实验3程序: \n");
    printf(" 1. 生成数据记录文件（使用二维数组存储）\n");
    printf(" 2. 生成数据记录文件（使用结构体数组存储）\n");
    printf("请输入您要执行的功能的编号\n");
    scanf("%d", &way); // 判断需要的存储方式
    clearstdin();
    if (way == 1) // 第一种方式，使用二维数组存储
    {
        int num[500][3] = { 0 };
        creat_randnumber_byshuzu(fp, data, num); // 数组生成随机数
        if ((fp = fopen(temp, "w+")) == NULL)
        {
            printf("创建文件失败\n");
            return 0;
        }
    }
}
```

图 1-2-2-7 Lab3 writenum 函数第一部分

选择“1.生成数据记录文件（使用二维数组存储）”，则进入 creat_randnumber_byshuzu 函数，进行随机数写入，最后打开文件进行数据写入。

选择“2.生成数据记录文件（使用结构体数组存储）”，先进行动态内存分配，之后进入 creat_randnumber_byjiegou 函数，进行随机数写入，最后打开文件进行数据写入。

```
fprintf(fp, "%d\n", data->number);
for (int i = 0; i < data->number; i++) // 输入到文件里
{
    fprintf(fp, "%d %d %d\n", num[i][0], num[i][1], num[i][2]);
    printf("%d %d %d\n", num[i][0], num[i][1], num[i][2]);
}
fclose(fp); // 关闭文件
}

if (way == 2) // 第二种方式，使用结构体数组存储
{
    DATAITEM* number; // 用动态规划划分结构体的空间
    number = (DATAITEM*)malloc(data->number * sizeof(DATAITEM));

    creat_randnumber_byjiegou(fp, data, number); // 用结构体存储数据内容
    if ((fp = fopen(temp, "w")) == NULL)
        printf("创建文件失败\n");
}

fprintf(fp, "%d\n", data->number);
for (int i = 0; i < data->number; i++) // 将随机数传输到文件里
{
    fprintf(fp, "%d %d %d\n", number[i].item1, number[i].item2, number[i].item3);
    printf("%d %d %d\n", number[i].item1, number[i].item2, number[i].item3);
}

fclose(fp); // 关闭文件夹
free(number); // 释放内存
```

图 1-2-2-8 Lab3 writenum 函数第二部分

7. void creat_randnumber_byshuzu(FILE* fp, CONF* data, int num[][3])

此函数支撑实验 3 中，用户选择使用二维数组存储，生成随机数对此进行处理。

先按照规定的数值范围进行划分，当相等的时候再次生成，最后存入数组。

```
/*
 * 函数名称: creat_randnumber_byshuzu
 * 函数功能: 将随机数生成后储存到数组里
 * 输入参数: CONF* data: 结构体指针data;a[][3]
 * 返回值: int(无实际意义)
 * 版本信息: create by Lifeng Zhang, 2023-04-17
 */
void creat_randnumber_byshuzu(FILE* fp, CONF* data, int num[][3])
{
    int i, temp1 = 0, temp2 = 0, temp3 = 0;
    for (i = 0; i < data->number; i++)
    {
        // 按照规定的数值范围进行划分
        temp1 = random((data->maxvalue1 - data->minvalue1) + data->minvalue1);
        temp2 = random((data->maxvalue1 - data->minvalue1) + data->minvalue1);

        if (temp1 == temp2) // 当相等的时候再次生成
            temp2 = random((data->maxvalue1 - data->minvalue1) + data->minvalue1);
        else
            temp3 = random((data->maxvalue2 - data->minvalue2) + data->minvalue2);

        // 存入随机数
        num[i][0] = temp1;
        num[i][1] = temp2;
        num[i][2] = temp3;
    }
}
```

图 1-2-2-9 Lab3 creat_randnumber_byshuzu 函数

8. void creat_randnumber_byjiegou(CONF* data, int num[][3])

此函数支撑实验 3 中，用户选择使用结构体数组存储，生成随机数对此进行处理。

先按照规定的数值范围进行划分，当相等的时候再次生成，最后存入数组。

```
*/  
* 函数名称: creat_randnumber_byjiegou  
* 函数功能: 将动态随机生成的随机数生成后储存到结构体里  
* 输入参数: CONF* data: 结构体指针data; DATAITEM* item: 存储数据记录结构体  
* 返回值: int(无实际意义)  
* 版本信息: create by Lifeng Zhang, 2023-04-17  
*/  
int creat_randnumber_byjiegou(FILE* fp, CONF* data, DATAITEM* item)  
{  
    int i, temp1 = 0, temp2 = 0, temp3 = 0;  
    for (i = 0; i < data->number; i++)  
    {  
        // 按照预先的范围生成随机数  
        temp1 = random((data->maxvalue1 - data->minvalue1) + data->minvalue1);  
        temp2 = random((data->maxvalue1 - data->minvalue1) + data->minvalue1);  
        if (temp1 == temp2)  
            temp2 = random((data->maxvalue1 - data->minvalue1) + data->minvalue1);  
        else  
            temp3 = random((data->maxvalue2 - data->minvalue2) + data->minvalue2);  
        item[i].item1 = temp1;  
        item[i].item2 = temp2;  
        item[i].item3 = temp3;  
    }  
    return 0;  
}
```

图 1-2-2-10 Lab3 creat_randnumber_byjiegou 函数

1.2.3 问题 c

.....请同学们思考程序是如何判断用户输入的数据记录条数是否合法的，请在实验报告中论述你的判断逻辑并将代码片段的截图粘贴在实验报告中。.....

答：

进入程序后，判断文件是否以“.txt”结尾。不是结束程序，否则继续判断。

之后判断文件是否以“\”开头。是结束程序，否则继续判断。

```
// 实验要求文件以.txt结尾
if (str[len - 1] != 't' && str[len - 2] != 'x' && str[len - 3] != 't' && str[len - 4] != '.')
    return 0;

if (*str == '\\') // 不能以“\”开头
    return 0;
```

图 1-2-3-1 判断流程图 1

之后判断文件是否以“\”或“/”结尾。是结束程序，否则继续判断。

```
// 不能以“\”或“/”结尾，若以此结尾则认为文件名为空
char* index1 = strrchr(str, '\\');
char* index2 = strrchr(str, '/');
if (index1 - str + 1 == strlen(str) || index2 - str + 1 == strlen(str))
    return 0;
```

图 1-2-3-2 判断流程图 2

之后判断文件路径是绝对路径还是相对路径。

```
// 判断输入路径为绝对路径还是相对路径
char* p1 = strstr(str, ":\\");
char* p2 = strstr(str, ":/");

if (p1 || p2) // 输入的是绝对路径
```

图 1-2-3-3 判断流程图 3

如果是绝对路径，使用_splitpath 拆分后，判断文件路径、文件名是否合法。
不合法结束程序，反之存储路径。

```
// 四个参数分别代表四个需要从原始文件路径中截取的字符串，有驱动器盘符(disk)，中间的路径 (dir)，文件名(fname)，和后缀名 (ext)
_splitpath(str, disk, dir, fname, ext);

// 判断disk在电脑中是否存在
if (access(disk, 0) != 0)
{
    printf("%s is no exist! disk在电脑中不存在\n", disk);
    return 0;
}

// 判断路径中是否有非法字符
if (strstr(dir, "*") || strstr(dir, "?") || strstr(dir, "<") || strstr(dir, ">") || strstr(dir, "|") || strstr(dir, ":"))
    return 0;
if (strstr(fname, "*") || strstr(fname, "?") || strstr(fname, "<") || strstr(fname, ">") || strstr(fname, "|") || strstr(fname, ":"))
    return 0;

// 储存路径、文件名
// void _makepath( const char *path, char *drive, char *dir, char *fname, char *ext); #include <direct.h>
// 第一个是要储存的完整的文件名路径
// 四个参数分别代表四个需要从结合字符串，有驱动器盘符(disk)，中间的路径 (dir)，文件名(fname)，和后缀名 (ext)
_makepath(way, disk, dir, NULL, NULL);
_makepath(file, NULL, NULL, fname, ext);

strcpy(data->filesavepath, way);
strcpy(data->filename, file);

if (access(data->filesavepath, 0) == -1) //路径不存在，创建新路径
    mkdir(data->filesavepath);
```

图 1-2-3-3 判断流程图 4

如果是相对路径，判断字符串是否合法。不合法结束程序，反之存储路径。

```
else if (p1 == NULL || p2 == NULL) //仅仅是文件名
{
    if (strstr(str, "*") || strstr(str, "?") || strstr(str, "<") || strstr(str, ">") || strstr(str, "|") || strstr(str, ":"))
        return 0;

    strcpy(data->filename, str);
}

else //相对路径
{
    if (strstr(str, "*") || strstr(str, "?") || strstr(str, "<") || strstr(str, ">") || strstr(str, "|") || strstr(str, ":"))
        return 0;

    // 储存路径、文件名
    strcpy(data->filename, str);
    strcpy(data->filesavepath, "\\0");
}
```

图 1-2-3-3 判断流程图 5

1.2.4 问题 d

若要将图 3-1 中“根据配置信息结构体变量中的信息生成数据文件”的流程封装为一个函数，函数的输入参数和返回值应该是什么？请在实验报告中给出该函数的函数声明，并对函数的输入参数及返回值加以解释说明。

答：

函数声明为：int creat_randnumber_byjiegou(CONF*, DATAITEM*);

输入参数为：

CONF* data: 结构体指针 data。

DATAITEM* item: 存储数据记录结构体。

返回值为：int（无实际意义）

```
*
* 函数名称: creat_randnumber_byjiegou
* 函数功能: 将动态随机生成的随机数生成后储存到结构体里
* 输入参数: CONF* data: 结构体指针 data; DATAITEM* item: 存储数据记录结构体
* 返回值: int(无实际意义)
* 版本信息: create by Lifeng Zhang, 2023-04-17
*/
nt creat_randnumber_byjiegou(CONF* data, DATAITEM* item)

int i, temp1 = 0, temp2 = 0, temp3 = 0;
for (i = 0; i < data->number; i++)
{
    // 按照预先的范围生成随机数
    temp1 = random((data->maxvalue1 - data->minvalue1) + data->minvalue1);
    temp2 = random((data->maxvalue1 - data->minvalue1) + data->minvalue1);
    if (temp1 == temp2)
        temp2 = random((data->maxvalue1 - data->minvalue1) + data->minvalue1);
    else
        temp3 = random((data->maxvalue2 - data->minvalue2) + data->minvalue2);
    item[i].item1 = temp1;
    item[i].item2 = temp2;
    item[i].item3 = temp3;
}
return 0;
```

图 1-2-4-1 Lab3 creat_randnumber_byjiegou 函数

1.2.5 问题 e

如何封装“获取一个指定范围内的随机整数的函数”（图 3-3 红色部分），它的输入参数和返回值应该是什么，请在实验报告中给出你实现的该函数的声明截图，对函数输入参数和返回值加以说明。

答：

此函数功能在 creat_randnumber_byjiegou 函数中写入。

函数参数返回值、声明、输入参数同问题 d。此处不再赘述。

```
// 按照预先的范围生成随机数
temp1 = random((data->maxvalue1 - data->minvalue1) + data->minvalue1);
temp2 = random((data->maxvalue1 - data->minvalue1) + data->minvalue1);
if (temp1 == temp2)
    temp2 = random((data->maxvalue1 - data->minvalue1) + data->minvalue1);
else
    temp3 = random((data->maxvalue2 - data->minvalue2) + data->minvalue2);
item[i].item1 = temp1;
item[i].item2 = temp2;
item[i].item3 = temp3;
```

图 1-2-5-1 实现指定范围函数部分（Lab3 creat_randnumber_byjiegou 函数）

1.2.6 问题 f

同学们统计你所编写代码的代码行数，尝试区分哪些代码是用于保障程序的健壮性，哪些代码是用来实现程序的功能，比较两部分代码数量，看二者的比例是多少，在实验报告中结合以上数据分析说明你对程序健壮性的理解。

答：

程序的健壮性是程序遇到相关特殊情况处理问题的一种能力。这种能力是指，程序在遇到无效输入、错误输入、某些外在压力的情况下，程序能够正确解决问题的程度。

针对于 Lab3_fun.cpp 文件的函数中，共计 521 行。删去注释共有 371 行。

A screenshot of a code editor showing three lines of C++ code. Line 519 contains the statement `data->number = a`. Line 520 contains the statement `return 1;`. Line 521 contains a closing curly brace `}`. The lines are numbered on the left side of the editor.

图 1-2-6-1 Lab3_fun.cpp 文件行数

其中，193 行是实现程序的功能，178 行是保障程序的功能。

比例为：（保障程序行数/实现程序的功能）=0.922。

依次类推，当程序运行正常，保障程序越多，则解决相关特殊情况能力越强，即程序健壮性越强。

1.2.7 问题 g

在程序中，配置信息结构体变量起到了什么作用，请同学们在实验报告中就你对这一问题的理解加以说明。

答：

配置信息结构体变量可以认为是每个函数之间的桥梁，通过数据结构体的变量，规范约束每一个函数中的内容，并且提供函数内容。这样子可以十分方便的使用函数数据，不必每次都写入特别长的形式参数传入其中。

```
char user_filename[MAX_STR_LEN];

int run(int argc, char* argv[]);

void ini_read(CONF* data);

void create_1(FILE*, int*, char* argv[], CONF*);
void create_2(FILE*, int*, char* argv[], CONF*);
void create_3(FILE*, int*, char* argv[], CONF*);

void get_filename(CONF*);
void get_filenum(CONF*);
int check_seper_file(char*, CONF*);
int writenum(FILE*, CONF*, char*);
void creat_randnumber_byshuzu(FILE*, CONF*, int a[][3]);
int creat_randnumber_byjiegou(CONF*, DATAITEM*);

int clearstdin();
int check_num(CONF*, char*);
```

图 1-2-6-1 Lab3_fun.h 文件 含有 CONF*情况

这样子使用，可以随时对结构体内的配置参数进行使用与更改，减少了许多形式参数与指针的传递，使整个程序更加结构化，规范化。

1.2.8 问题 h

在流程图 3-1、3-2、3-3 中，有哪些流程可以封装为子函数，请同学们加以分析，并尝试将一些流程封装为函数，给出函数的声明并解释说明函数功能及输入输出参数。

答：（以下流程相关截图请看报告中“1.2.1 问题 a”“1.2.2 问题 b”）

以下流程封装为了子函数：

1. 文件名是否合法：

```
int check_seper_file(char* str, CONF* data)
```

2. 随机生成条数输入是否合法：

```
void get_filenum(CONF* data)
```

3. 将用户输入的记录条数信息写入配置信息结构体变量/数组

```
void get_filenum(CONF* data)
```

```
void get_filename(CONF* data)
```

```
int writenum(FILE* fp, CONF* data, char* temp)
```

```
void creat_randnumber_byshuzu(FILE* fp, CONF* data, int num[][3])
```

```
int creat_randnumber_byjiegou(CONF* data, DATAITEM* item)
```

4. 为结构体变量各个分量初始化默认值

```
void ini_read(CONF* data)
```

5. 判断命令行参数是否为数值

6. 打印并显示菜单

```
int check_num(CONF* data, char* str)
```

实验总结

2 遇到的问题及解决办法

2.1 程序与调试相关问题

2.1.1 指针不清晰——LNK1120、LNK2019 报错

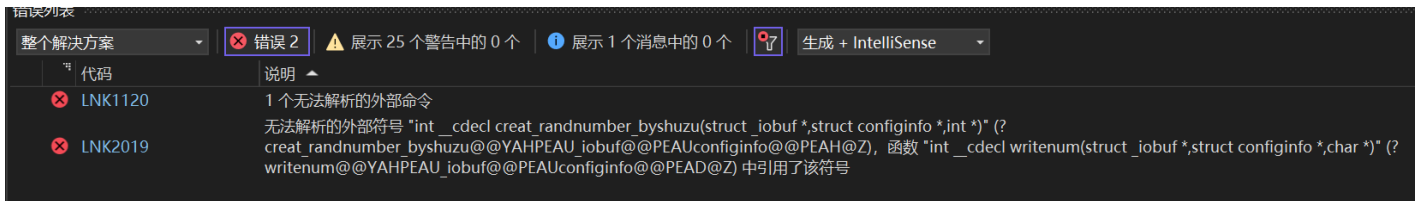


图 2-1-1 LNK1120、LNK2019 报错

编写程序后进行调试时，发现部分时候程序出现错误，并下方报错

LNK1120、LNK2019。

查询相关网页解释如下：

原因

Add 函数声明如下：

```
int Add(void const* p , int timeout = 0);
```

实现接口如下：

```
int Add(void const* p) {}
```

说明函数声明跟实现接口不一致，导致的问题

修改如下： `int Add(void const* p, int timeout) {}`^[1]

经过长时间排查得出以下结论：

声名中的 `void creat_randnumber_byshuzu(FILE*,CONF*, int*)` 中的 `int*` 是指向整数的指针类型，而定义中的 `int* num[][3]` 是二维指针数组类型。两者冲突导致程序报错。

```
void ini_read(CONF* data);

void create_1(FILE*, int*, char* argv[], CONF*);
void create_2(FILE*, int*, char* argv[], CONF*);
void create_3(FILE*, int*, char* argv[], CONF*);

void get_filename(CONF*);
void get_filenum(CONF*);
int check_seper_file(char*, CONF*);
int writenum(FILE*, CONF*, char*);
void creat_randnumber_byshuzu(FILE*, CONF*, int*);
int creat_randnumber_byjiegou(FILE*, CONF*, DATAITEM*);

int clearstdin();
int check_num(char*);
```

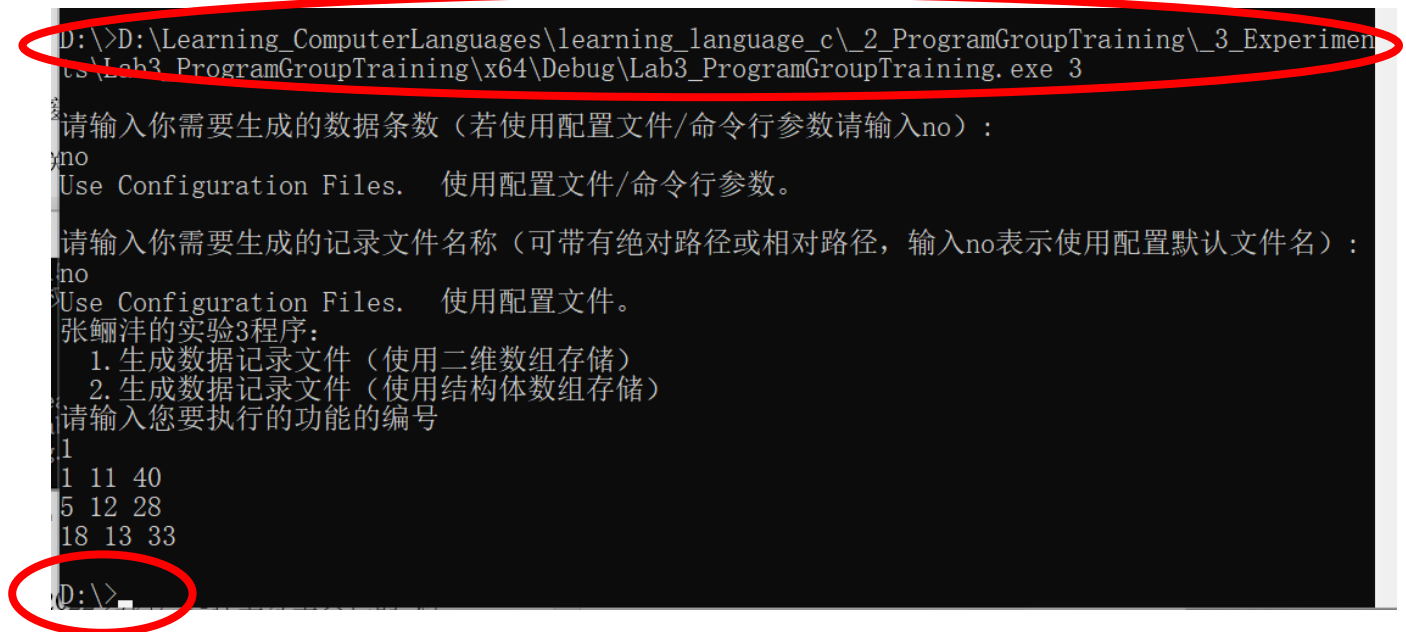
图 2-1-2 函数声明

```
/*
 * 函数名称: creat_randnumber_byshuzu
 * 函数功能: 将随机数生成后储存到数组里
 * 输入参数: CONF* data: 结构体指针data;a[][3]
 * 返回值: int(无实际意义)
 * 版本信息: create by Lifeng Zhang, 2023-04-17
 */
void creat_randnumber_byshuzu(FILE* fp, CONF* data, int* num[][3])
{
    int i, temp1 = 0, temp2 = 0, temp3 = 0;
    for (i = 0; i < data->number; i++)
    {
        // 按照规定的数值范围进行划分
    }
}
```

图 2-1-3 函数定义段

2.1.2 cmd 窗口与工作目录

进行程序相关测试时，在下面情况时，文件生成总是不在理想的位置。



```
D:\>D:\Learning_ComputerLanguages\learning_language_c\2_ProgramGroupTraining\_3_Experiments\Lab3_ProgramGroupTraining\x64\Debug\Lab3_ProgramGroupTraining.exe 3

请输入你需要生成的数据条数（若使用配置文件/命令行参数请输入no）：
no
Use Configuration Files. 使用配置文件/命令行参数。

请输入你需要生成的记录文件名称（可带有绝对路径或相对路径，输入no表示使用配置默认文件名）：
no
Use Configuration Files. 使用配置文件。

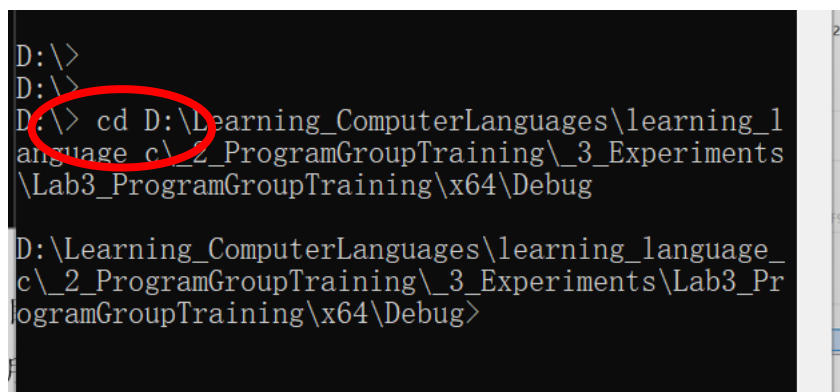
张颢沅的实验3程序：
1. 生成数据记录文件（使用二维数组存储）
2. 生成数据记录文件（使用结构体数组存储）
请输入您要执行的功能的编号
1
1 11 40
5 12 28
18 13 33
D:\>
```

图 2-1-4 cmd 运行窗口 1

此时生成文件并未在 exe 文件所在目录，而是在 D 盘下的直接目录里。

通过思考发现，文件的工作目录是在 D 盘，而不是在 exe 文件所在目录。

在 cmd 窗口，输入“cd+地址”，如下图：



```
D:\>
D:\>
D:\>cd D:\Learning_ComputerLanguages\learning_language_c\2_ProgramGroupTraining\_3_Experiments\Lab3_ProgramGroupTraining\x64\Debug
D:\Learning_ComputerLanguages\learning_language_c\2_ProgramGroupTraining\_3_Experiments\Lab3_ProgramGroupTraining\x64\Debug>
```

图 2-1-5 cmd 运行窗口 2

此时，文件的工作目录是在 exe 文件所在目录了。这个方式也可以链接到其他文件夹。

3 实验的收获与心得

(1) 动态内存分配与结构体使用

本次实验是第一次结合动态内存分配与结构体，进行随机数生成。通过这次训练，体会到了动态内存分配的相关使用规则。

(2) 指针与函数接口

本次实验涉及较多的指针运用。本次程序设计中，发现对于指针的掌握不太好，出现了较多次的报错。后续需要在写程序的过程中需要继续训练。

(3) 绝对目录与相对目录

本次实验重点之一是分割绝对目录与相对目录。本次程序设计中，运用了 `_splitpath` 等函数对目录进行处理，对绝对目录与相对目录有了一定的了解。

本次实验受益良多，为后面实验 4、5 打下了坚实的基础。希望在以后的学习中会有更多的收获。

参考文献

[1] fengyuzaitu 无法解析的外部符号 `int __cdecl test(void const *,int)`解决过程
<https://blog.51cto.com/fengyuzaitu/2576625> .2020-12-29

相关附件：

附件 1: Lab3 文件夹。内含源文件代码与编译程序

附件 2: Lab3_仅代码 文件夹。内仅含源文件代码（即 `xxx.cpp`、`xxx.h` 文件）

附件 3: Lab3_程序流程图.jpg

附件 4: Lab3_运行程序.exe