

**程序设计分组训练实验四**

**实验报告**

作者:柯劲帆

学号: 21281280

日期:2022年5月22日

目录

[1. 概述 1](#_Toc104216648)

[1.1. 标识 1](#_Toc104216649)

[1.2. 范围 1](#_Toc104216650)

[2. 程序设计需求 1](#_Toc104216651)

[2.1. 功能需求 1](#_Toc104216652)

[3. 程序详细设计 3](#_Toc104216653)

[3.1. 功能详细设计 3](#_Toc104216654)

[3.2. 程序与外部程序协同设计 4](#_Toc104216655)

[3.3. 配置文件设计 4](#_Toc104216656)

[3.4. 程序工程文件组织设计 5](#_Toc104216657)

[3.4.1. 程序源文件设计 5](#_Toc104216658)

[3.5. 内存数据数据模型设计 5](#_Toc104216659)

[3.6. 函数接口设计 6](#_Toc104216660)

[3.7. 函数详细设计 10](#_Toc104216661)

[3.7.1. 程序入口函数main 10](#_Toc104216662)

[3.7.2. 程序主函数run 11](#_Toc104216663)

[3.7.3. ……（根据你的程序补充后续函数详细设计内容） 12](#_Toc104216664)

[3.8. 程序交互设计 12](#_Toc104216665)

[3.8.1. 系统主菜单设计 12](#_Toc104216666)

[3.8.2. 修改配置文件子菜单设计 13](#_Toc104216667)

[3.8.3. 菜单循环展示设计 13](#_Toc104216668)

[3.8.4. 用户交互提示信息设计 15](#_Toc104216669)

[3.8.5. 数据记录显示输出交互设计 16](#_Toc104216670)

# 概述

## 标识

文档名称：程序设计分组训练实验4—程序设计说明书

文档编号：Lab4-21281280

## 范围

本文档适用于《程序设计分组训练》课程，为课程实验4环节的交付物。文档用于描述程序设计分组训练课程中实验4所编写程序的设计方案。文档阅读对象为本课程授课教师及本课堂同学。

# 程序设计需求

## 功能需求

根据《程序设计分组训练》课程实验4要求，待编写的程序需要实现以下功能：

1）设计程序读取实验4中输出的数据文件，程序能够根据文件扩展名自动识别是文本形式的数据记录文件还是二进制形式的数据记录文件；

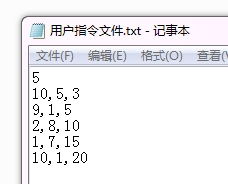


图2-1文本形式的数据记录文件存储格式样例

以文本形式存储的数据记录文件为例，其中文件的第一行用于存储文件的记录数，如图2-1中，表示数据文件中总共存储了5条数据记录；从文件第二行开始，逐条存储数据记录文件，数据记录文件是一个三元组<元素1,元素2,元素3>，每个元素为一个int值，元素与元素之间用“,”（英文逗号）进行分隔；

2）设计程序读取实验4中输出的数据文件，将文件中的数据记录文件存入以下四种不同的数据结构中：

* 二维数组：N行3列的二维数组，每行存储一条数据记录，数组空间根据数据记录文件中的记录数动态申请；
* 结构体数组：长度为N的结构体数组，每一个数组元素为一个结构体，记录一条数据记录信息，数组空间根据数据记录文件中的记录数动态申请。请自行设计结构体声明；
* 指针数组：长度为N的指针数组，每一个数组元素为一个指向结构体的指针，每一个结构体变量存储数据记录文件中的一条数据记录，指针数组及数组元素和指向的结构体变量空间均由malloc函数动态生成；
* 链表：包含头结点的单向链表，头结点中存储数据记录个数及指向第一个数据结点的指针，数据结点存储一条数据记录信息及指向下一个数据结点的指针，链表结构体请自行定义，并采用malloc函数动态生成；

3）设计实现以下功能：

* 能够以外部程序的方式对实验3的程序进行调用，用实验3的exe程序生成数据记录文件，用实验4程序读取并显示数据记录文件内容；
* 尝试设计合适的文件目录关系，将实验3编译生成的exe文件与实验4有机整合起来，使得实验4最终生成的exe能够与实验3编译生成的exe有效协同开展工作；
* 程序通过菜单方式向用户提供功能；
* 程序运行后需要能够循环显示功能菜单，用户选择的功能序号进行相应处理，处理完成后程序再次显示功能菜单，等待用户选择其他功能，只有用户选择0号功能时，才退出程序。
* 实验4程序提供两种工作模式：自动模式与交互模式，各模式下程序工作过程如下：
  + 自动模式：在该模式下，用户选择要执行的功能后，调用实验3时，实验3以其conf.ini里面的配置信息作为参数，直接在默认的存储位置，以默认的文件名生成数据记录文件；
  + 交互模式：在该模式下，用户选择要执行的功能后，由程序提供与用户的交互功能，用户输入文件的存储位置、文件名和记录条数参数，结合conf.ini里的其他配置信息，生成数据记录文件；
* 提供加载配置文件功能，配置文件用于存储实验4的工作模式；
* 在功能菜单所列的功能中，除了1、14、0号功能外，用户选择其它功能时当实验4将数据记录文件读取到内存容器中之后，程序需提供数据展示功能，由专门的数据展示函数将内存容器中的数据记录读出，显示到演示屏上，供用户查看；
* 在功能菜单所列的功能中，针对10、11、12号功能，对于采用二维数组，结构体数据，指向结构体的指针数组三种存储方式，采用qsort函数对从数据记录文件中读取出来的数据进行排序，排序规则是按照数据记录三元组中的第三个元素的大小，从小到大排列；
* 在上述功能菜单所列的功能中，针对13号功能，采用冒泡排序或快速排序方法，设计实现对第四种链表存储方式的排序，排序规则与其它三种一致；
* 实验4程序需要提供对实验3的配置文件conf.ini的修改功能（功能选项14）；
* 实验4程序需要提供对实验4程序的配置文件中工作模式参数的修改功能；

4）程序设计时请遵守以下要求：

* 实验4主函数要求由一条语句实现，实验4的所有功能均实现在run( )函数内；
* 以多文件工程的方式组织实验4程序的开发；

# 程序详细设计

## 功能详细设计

根据《程度设计分组训练》课程实验4的题目要求，将实验4程序功能细化如下：

1）外部程序调用功能：外部程序调用功能主要实现对实验3编译好的exe程序进行调用，通过调用实验3程序生成数据记录文件的功能。调用外部文件有两种调用方式：

* 自动模式：在该模式下，调用实验3程序时，实验3以其conf.ini里面的配置信息作为参数，直接在默认的存储位置，以默认的文件名生成数据记录文件，文件记录条数随机生成；
* 交互模式：在该模式下，实验4程序提供与用户的交互功能，用户输入文件的存储位置、文件名和记录条数参数，实验4程序以命令行参数的形式发送给实验3程序，生成数据记录文件；

2）数据加载功能：数据加载功能主要实现将外部数据记录文件的数据记录读出并存入到四种不同的存储结构中去的功能。数据记录文件的位置指定有两种程序调用方式：

* 自动模式：实验4程序到约定好的数据记录文件存储位置找到数据记录文件并进行读取加载；
* 交互模式：实验4程序与用户进行交互，由用户输入文件的存储位置和文件名信息，实验4程序到用户指定的位置打开用户指定的文件，进行读取加载；

3）数据展示功能：数据展示功能主要实现将4中不同存储结构中存储的数据记录显示到控制台界面的功能；

4）数据排序功能：数据排序功能能够将读取出的数据记录按指定顺序规则排序；

4）菜单展示及功能调用功能：菜单展示与功能调用功能主要实现程序启动后在界面显示系统功能菜单并根据用户输入的功能选项调用相应的功能；

5）配置文件conf.ini修改功能：主要实现对实验3的配置文件conf.ini中各个配置参数进行修改以及对实验4的工作模式的修改功能。

## 程序与外部程序协同设计

根据《程度设计分组训练》课程实验4的题目要求，实验4程序需要调用已编译好的实验3的exe程序，并读取实验4程序生成的数据记录文件，为了实现实验4程序与实验3程序的有效配置，对实验3程序、实验4程序及生成的数据记录文件存储位置作以下约定：

1. 当处于自动模式下，实验3生成的数据记录文件存储于与实验3程序同级目录的DataSet子目录下，数据记录文件的默认名称为DataFile.txt；
2. 当处于交互模式下，实验3生成的数据记录文件存储于用户指定的合法目录中，数据记录文件名也由用户指定；
3. 当处于自动模式下，实验4读取数据记录文件时，从与实验3程序同级目录的DataSet子目录下，打开名称叫DataFile.txt的文件进行数据加载；
4. 当处于交互模式下，实验4根据用户指定的目录和文件名进行文件读取和数据加载；
5. 实验3程序与实验4程序存放于同一级目录下；

实验4与外部程序协同关系示意图如图3-1所示。



图3-1 实验4与实验3协同关系示意图

## 配置文件设计

根据《程度设计分组训练》课程实验4的题目要求，实验4程序需要读取配置文件，从中获取实验4程序的工作模式。对实验4配置文件设计如下：

* 文件名称：conf.ini
* 文件存储位置：同一文件夹下
* 文件内容说明：无（工作模式由用户自己选择）

## 程序工程文件组织设计

### 程序源文件设计

根据《程度设计分组训练》课程实验4的题目要求，将整个实验4的功能划分为七块，由7个源文件和7个头文件实现，具体划分形式如表3-1所示。

表3-1 程序设计分组训练实验4程序模块划分说明表

|  |  |  |
| --- | --- | --- |
| 模块名称 | 文件名称 | 文件说明 |
| 结构体及常量声明模块 | Lab4\_data.h | 存放实验5程序所需的结构体声明、常量声明及全局变量声明 |
| 主模块 | Lab4.cpp | 程序入口文件，存放整个工程的main函数 |
| Lab4\_main.c | 程序主控文件，存放run函数及其他需要配套的子函数 |
| Lab4\_main.h | 主控文件 的头文件，存放主控文件的函数说明及程序中主要结构体和全局变量声明 |
| 外部程序调用模块 | Lab4\_call.c | 存放调用实验4程序生成数据记录文件的函数 |
| Lab4\_call.h | 存放调用实验4程序生成数据记录文件函数的声明 |
| 数据加载模块 | Lab4\_loadfile.c | 存放数据读取、数据加载相关功能所需的函数 |
| Lab4\_loadfile.h | 存放数据加载相关函数的函数声明 |
| 数据显示模块 | Lab4\_view.c | 存放将存储在内存容器中的数据记录显示输出到屏幕上功能的相关函数 |
| Lab4\_view.h | 存放数据显示模块相关函数的函数声明 |
| 系统配置模块 | Lab4\_conf.c | 存放与修改配置文件相关的功能函数 |
| Lab4\_conf.h | 存放修改配置文件相关功能函数的函数声明 |
| 排序模块 | Lab4\_sort.c | 存放与排序功能相关的功能函数 |
| Lab4\_sort.h | 存放与排序功能相关功能函数的函数声明 |
| 工具模块 | Lab4\_tools.c | 存储模块常用的而主函数不调用的工具函数 |
| Lab4\_tools.h | 存储模块常用的而主函数不调用的工具函数声明 |

## 内存数据数据模型设计

本程序需要处理的数据主要是数据记录文件中的记录数据，每条记录数据为一个由3个整型值组成的三元组（见2.1节）。在程序中，采用一个四种存储结构来存储数据记录文件中的记录数据，

1.二维数组存储方式

二维数组为N行3列，动态申请空间，二维数组声明如下：

int(\*data)[3] = (int(\*)[3])malloc(sizeof(int) \* 3 \* dataNum);

2.结构体数组存储方式

DATAITEM\* data = (DATAITEM\*)malloc(sizeof(DATAITEM) \* dataNum);

3.指针数组存储方式

DATAITEM\*\* data = (DATAITEM\*\*)malloc(sizeof(DATAITEM\*) \* dataNum);

4.链表存储方式

NODE\* head = (NODE\*)malloc(sizeof(NODE));

程序中需涉及的其他常量和变量如表3-2所示。

表3-2 程序关键常量、变量一览表

|  |  |  |
| --- | --- | --- |
| 声明 | 常量/变量 | 说明 |
| CONF ini | 变量 | 创建结构体变量存储生成数据记录文件的限制参数以及数据记录条数 |
| char filename[MAX\_STR\_LEN \* 2] | 变量 | 数据记录文件路径和文件名 |

## 函数接口设计

整个程序共设计函数45个，函数功能及接口设计如表3-3所示。

表3-3 程序函数设计一览表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 函数名称 | 所属文件 | 输入参数说明 | 返回值说明 | 函数说明 |
| main | Lab4.cpp | 命令行参数 | main函数返回0 | 程序入口函数 |
| run | Lab4\_main.c | 命令行参数 | 无 | 程序主函数 |
| self\_check\_and\_get\_configinfo | Lab4\_loadfile.c | 生成数据记录文件的限制参数指针CONF\* pIni，数据记录文件路径和文件名filename | 无 | 检查Lab3.exe是否可用，并从conf.ini中读取生成数据记录文件的限制参数 |
| change\_Filename\_to\_absolute | Lab4\_loadfile.c | 记录文件名的地址filename | 执行情况error（1-路径无误，0-路径有误） | 将文件路径转化为绝对路径 |
| create\_path | Lab4\_loadfile.c | 记录文件名的地址filename | 无 | 如果路径未被创建则创建路径 |
| check\_filename | Lab4\_loadfile.c | 记录文件名的地址filename | 文件路径是否有误error（1-路径无误，0-路径有误） | 检查文件名是否合法 |
| fget\_file\_path\_and\_name | Lab4\_loadfile.c | 用于存放数据记录文件的存储目录或文件名的字符串s，读取目标文件fp | 无 | 从文件中读取用于存放数据记录文件的存储目录或文件名的字符串 |
| fget\_num | Lab4\_loadfile.c | 读取目标文件fp | 生成数据记录文件的限制参数num | 从文件中读取生成数据记录文件的限制参数并转化为整型 |
| get\_dataNum | Lab4\_loadfile.c | 数据记录条数信息的状态的指针pDataNumStatus，生成数据记录文件的限制参数指针CONF\* pIni | 无 | 获取数据记录条数 |
| get\_filename | Lab4\_loadfile.c | 数据记录文件路径信息的状态的指针pFilenameStatus，数据记录文件路径和文件名filename | 无 | 获取文件路径 |
| store\_data\_in\_2d\_array | Lab4\_loadfile.c | 二维数组地址int(\*data)[3]，文件指针fp，数据记录条数dataNum | 无 | 从文件读取内容进二维数组 |
| store\_data\_in\_struct\_array | Lab4\_loadfile.c | 结构体数组地址DATAITEM\* data，文件指针fp，数据记录条数dataNum | 无 | 从文件读取内容进结构体数组 |
| store\_data\_in\_point\_array | Lab4\_loadfile.c | 结构体数组地址DATAITEM\*\* data，文件指针fp，数据记录条数dataNum | 无 | 从文件读取内容进指针数组 |
| store\_data\_in\_link\_list | Lab4\_loadfile.c | 单链表头节点（value为空）NODE\* head，文件指针fp，数据记录条数dataNum | 无 | 从文件读取内容进单链表 |
| get\_dataNum\_from\_file | Lab4\_loadfile.c | 文件指针fp | 数据记录条数（int） | 从文件开头读取数据记录条数 |
| call\_program | Lab4\_call.c | 是否是自动模式autoMode，生成数据记录文件的限制参数指针CONF\* pIni，数据记录文件路径和文件名filename，数据记录文件路径信息的状态的指针pFilenameStatus，数据记录条数信息的状态的指针pDataNumStatus | 无 | 请求用户确认数据是否无误并调用外部程序 |
| modify\_configinfo | Lab4\_conf.c | 生成数据记录文件的限制参数指针CONF\* pIni | 无 | 修改配置文件 |
| cmp\_for\_2d\_array\_qsort | Lab4\_sort.c | 比较的两个二维数组的地址const void\*\* a、const void\*\* b | <0（不进行置换） / >0（进行置换） / 0（不进行置换） | 为给二维数组使用qsort编写的cmp函数 |
| cmp\_reverse\_for\_2d\_array\_qsort | Lab4\_sort.c | 比较的两个二维数组的地址const void\*\* a、const void\*\* b | <0(不进行置换) / >0(进行置换) / 0(不进行置换) | 为给二维数组使用qsort编写的cmp函数（逆序） |
| cmp\_for\_struct\_array\_qsort | Lab4\_sort.c | 比较的两个一维数组的地址const void\* a、const void\* b | <0(不进行置换) / >0(进行置换) / 0(不进行置换) | 为给结构体数组使用qsort编写的cmp函数 |
| cmp\_reverse\_for\_struct\_array\_qsort | Lab4\_sort.c | 比较的两个一维数组的地址const void\* a、const void\* b | <0(不进行置换) / >0(进行置换) / 0(不进行置换) | 为给结构体数组使用qsort编写的cmp函数（逆序） |
| cmp\_for\_point\_array\_qsort | Lab4\_sort.c | 比较的两个二维数组的地址const void\*\* a、const void\*\* b | <0(不进行置换) / >0(进行置换) / 0(不进行置换) | 为给指针数组使用qsort编写的cmp函数 |
| cmp\_reverse\_for\_point\_array\_qsort | Lab4\_sort.c | 比较的两个二维数组的地址const void\*\* a、const void\*\* b | <0(不进行置换) / >0(进行置换) / 0(不进行置换) | 为给指针数组使用qsort编写的cmp函数（逆序） |
| bubble\_for\_link\_list | Lab4\_sort.c | 单链表头节点（value为空）NODE\* head，链表长度dataNum，是否逆序reverse | 无 | 对单链表使用冒泡排序 |
| show\_menu\_root | Lab4\_view.c | 无 | 用户输入内容（int） | 显示选择用户/开发者模式的菜单并读取输入 |
| show\_menu\_user | Lab4\_view.c | 无 | 用户输入内容（int） | 显示用户模式的菜单并读取输入 |
| show\_menu\_developer | Lab4\_view.c | 无 | 用户输入内容（int） | 显示开发者模式的菜单并读取输入 |
| show\_menu\_ask\_if\_read\_file | Lab4\_view.c | 无 | 用户输入内容（1-Y；0-N） | 询问用户是否查看生成的数据记录文件 |
| show\_menu\_ask\_how\_to\_store | Lab4\_view.c | 无 | 用户输入内容（int） | 询问用户选择二维数组/结构体数组/指针数组/链表存储 |
| show\_menu\_timeCount | Lab4\_view.c | 无 | 用户输入内容（1-Y；0-N） | 开发者模式询问用户运行程序时是否计时 |
| show\_menu\_ask\_data\_status\_before\_sort | Lab4\_view.c | 无 | 用户输入内容（int） | 询问用户选择计时排序前数据预处理方式 |
| show\_menu\_ask\_sort\_method | Lab4\_view.c | 无 | 用户输入内容（int） | 询问用户选择排序方式 |
| print\_data\_in\_2d\_array | Lab4\_view.c | 二维数组地址int(\*data)[3]，数据记录条数dataNum | 无 | 打印二维数组内容 |
| print\_data\_in\_struct\_array | Lab4\_view.c | 结构体数组地址DATAITEM\* data，数据记录条数dataNum | 无 | 打印结构体数组内容 |
| print\_data\_in\_point\_array | Lab4\_view.c | 结构体数组地址DATAITEM\*\* data，数据记录条数dataNum | 无 | 打印指针数组内容 |
| print\_data\_in\_link\_list | Lab4\_view.c | 单链表头节点（value为空）NODE\* head | 无 | 打印链表内容 |
| show\_data\_in\_2d\_array | Lab4\_view.c | 数据记录文件路径和文件名filename，文件指针fp，用户/开发者模式rootMode | 无 | 排序并展示数据记录（二维数组） |
| show\_data\_in\_struct\_array | Lab4\_view.c | 数据记录文件路径和文件名filename，文件指针fp，用户/开发者模式rootMode | 无 | 排序并展示数据记录（结构体数组） |
| show\_data\_in\_struct\_array | Lab4\_view.c | 数据记录文件路径和文件名filename，文件指针fp，用户/开发者模式rootMode | 无 | 排序并展示数据记录（结构体数组） |
| show\_data\_in\_point\_array | Lab4\_view.c | 数据记录文件路径和文件名filename，文件指针fp，用户/开发者模式rootMode | 无 | 排序并展示数据记录（指针数组） |
| show\_data\_in\_link\_list | Lab4\_view.c | 数据记录文件路径和文件名filename，文件指针fp，用户/开发者模式rootMode | 无 | 排序并展示数据记录（链表） |
| str\_to\_int | Lab4\_tools.c | 存储数字信息的字符串numArr | 合法转化数字结果num，有非数字字符返回-2，超出范围返回-1 | 将numArr中的存储数据记录条数检查是否合法，合法转化为数字 |
| free\_data\_in\_point\_array | Lab4\_tools.c | 结构体数组地址DATAITEM\*\* data，数据记录条数dataNum | 无 | 释放指针数组的每个元素指向的空间 |
| free\_data\_in\_link\_list | Lab4\_tools.c | 单链表头节点（value为空）NODE\* head | 无 | 释放单链表的空间 |
| get\_time | Lab4\_tools.c | 存储当前时间的字符串 | 无 | 获取当前时间，输出格式：xx月xx日xx时xx分xx秒-xxxx年 |

## 函数详细设计

### 程序入口函数main

实验4入口函数要求只由一条语句实现，即调用实验4程序主函数，实验4的所有功能均实现在程序主函数内，实验4程序入口主函数代码如图3-2所示。

（请替换你的程序截图）

图3-2 程序设计分组训练实验4程序入口函数代码截图

### 程序主函数run

实验4主函数run用于实现实验4的所有程序功能，函数流程图如图3-3所示。



图3-3 实验4主程序流程图

实验4主函数run的输入参数及函数返回值说明如下：

* 输入参数：
  + argc：整 型，表示命令行参数个数；
  + argv：字符指针数组，每个数组元素指向一个命令行参数字符串；
* 函数返回值：无函数返回值。
* 函数功能：XXXXXXX……

### ……（根据你的程序补充后续函数详细设计内容）

（请补充函数声明、输入参数说明、返回值说明，并给出函数流程图，注意不要代码截图）

## 程序交互设计

### 系统主菜单设计

系统主菜单显示样式如下：

XXX的实验4程序：

1. 调用实验3程序生成记录文件

2.读取指定数据记录文件（二维数组存储方式）

3.读取指定数据记录文件（结构体数组存储方式）

4.读取指定数据记录文件（指针数组存储方式）

5.读取指定数据记录文件（链表存储方式）

6. 调用实验3生成数据记录文件，同时读取数据记录文件（二维数组方式存储）

7. 调用实验3生成数据记录文件，同时读取数据记录文件（结构体数组方式存储）

8. 调用实验3生成数据记录文件，同时读取数据记录文件（指针数组方式存储）

9. 调用实验3生成数据记录文件，同时读取数据记录文件（链表方式存储）

10. 调用实验3生成数据记录文件，同时读取数据记录文件并排序（二维数组方式存储）

11. 调用实验3生成数据记录文件，同时读取数据记录文件并排序（结构体数组方式存储）

12. 调用实验3生成数据记录文件，同时读取数据记录文件并排序（指针数组方式存储）

13. 调用实验3生成数据记录文件，同时读取数据记录文件并排序（链表方式存储）

14. 重新设置配置参数值

0. 退出

请输入您要执行的程序序号：

### 修改配置文件子菜单设计

当用户选择主菜单中的14号功能时，显示一个子菜单供用户选择要修改的配置参数，子菜单显示样式如下：

（给出你的子菜单显示样式）

### 菜单循环展示设计

程序运行后需要能够循环显示功能菜单，用户选择的功能序号进行相应处理，处理完成后程序再次显示功能菜单，等待用户选择其他功能，只有用户选择0号功能时，才退出程序。当用户选择14号程序功能时，进入配置文件修改子菜单，在子菜单中选择功能0会返回上级主菜单。程序菜单显示功能流程如图3-4所示；



图3-4 程序菜单显示功能流程图

程序实现伪代码如表3-4所示。

表3-4 程序菜单显示伪代码

|  |
| --- |
| **……**  **int is\_Run=1;**  **while(is\_Run)**  **{**  **在屏幕上打印菜单；**  **while(scanf(“%d”,&func\_index)!=1 && （func\_index < 0 || func\_index > 14)）**  **{**  **printf(“您选择的功能不正确，请重新输入\n”);**  **fflush(stdin);**  **}**  **switch(func\_index)**  **{**  **case 0:**  **is\_Run = 0;**  **break;**  **case 1:**  **调用函数完成功能1；**  **break;**  **case 2:**  **……**  **case 14:**  **while(1)**  **{**  **在屏幕上打印子菜单**  **while(scanf(“%d”,&subfunc\_index)!=1 && （subfunc\_index < 0 || subfunc\_index > 功能最大序号)）**  **{**  **printf(“您选择的功能不正确，请重新输入\n”);**  **fflush(stdin);**  **}**  **if(subfunc\_index == 0)**  **{**  **break;**  **}**  **else if(subfunc\_index == 1)**  **{**  **调用函数完成子功能1；**  **}**  **……**  **}**  **break;**  **}**  **}** |

### 用户交互提示信息设计

当实验4程序处于交互模式时，需要输出提示信息提示用户输入正确的参数，相关提示信息设计如下：

* 提示用户输入文件存储位置时的提示信息：

（你的提示信息的内容是什么，注意这里不要截图）

* 提示用户输入数据记录条数时的提示信息：

（你的提示信息的内容是什么，注意这里不要截图）

* 修改配置文件时提示用户输入新的文件存储目录的提示信息：

（你的提示信息的内容是什么，注意这里不要截图）

* 修改配置文件时提示用户输入新的文件名称的提示信息：

（你的提示信息的内容是什么，注意这里不要截图）

* 修改配置文件时提示用户输入第一、二个元素最大值的提示信息：

（你的提示信息的内容是什么，注意这里不要截图）

* 修改配置文件时提示用户输入第一、二个元素最小值的提示信息：

（你的提示信息的内容是什么，注意这里不要截图）

* 修改配置文件时提示用户输入第三个元素最大值的提示信息：

（你的提示信息的内容是什么，注意这里不要截图）

* 修改配置文件时提示用户输入第三个元素最小值的提示信息：

（你的提示信息的内容是什么，注意这里不要截图）

* 修改配置文件时提示用户输入记录条数最大值的提示信息：

（你的提示信息的内容是什么，注意这里不要截图）

* 修改配置文件时提示用户输入记录条数最小值的提示信息：

（你的提示信息的内容是什么，注意这里不要截图）

* 修改配置文件时提示用户输入实验5工作模式参数的提示信息：

（你的提示信息的内容是什么，注意这里不要截图）

### 数据记录显示输出交互设计

（论述在显示输出内存数据容器中的数据记录时以什么样的格式输出）

【注意】附录1所提供的软件设计报告模块主要用于说明程序设计的论述大纲，其文档格式方面（如字体、行距等）与实验1中《程序设计分组训练课程文档编写格式规范说明》要求可能会有出入、撰写文档时应《程序设计分组训练课程文档编写格式规范说明》的格式要求为准。