程序设计分组训练 实验 4
程序设计说明书

学期：2022-2023 第二学期
报告人：张鲴沣
学号：22281052
班级：计算机 2202 班
编制日期：2023 年 5 月 22 日

程序设计分组训练 实验 4
程序设计说明书

目录

1.概述

1.1. 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_ProgramGroupTraining_22281052

1.2. 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。

2.程序设计需求

2.1. 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，并进行排序，给出排序时间。

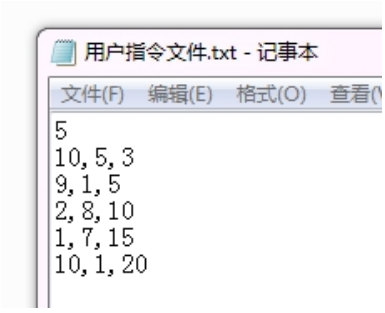

图 2-1 文本形式的数据记录文件存储格式样例

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，元素与元素之间用空格进行分隔；

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；

3.程序详细设计

3.1. 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；

5）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；

6）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。

3.2. 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。

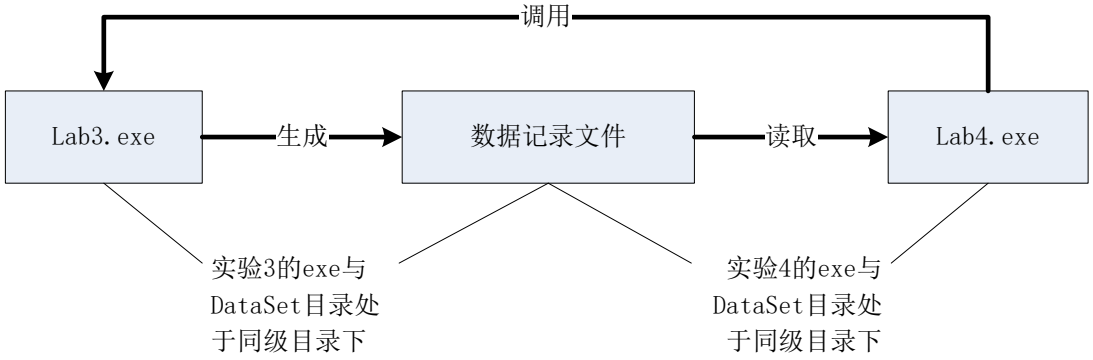

图 3-1 实验 4 与实验 3 协同关系示意图

3.3. 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：zlf_method.ini
- 文件存储位置：与 Lab4.cpp 所在文件夹下设的“set”文件夹
- 文件内容说明：通过对 zlf_method.ini 函数中的第一行内容，进行 1 或者 2 的记录。其中，1 代表是自动模式,2 代表交互模式

3.4. 程序工程文件组织设计

3.4.1. 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 4 程序所需的结构体声明、常量声明及全局变量声明
	Lab4_HeaderFile.h	存放实验 4 程序所需的系统库头文件与 Lab4_data.h
主模块	Lab4.cpp	程序入口文件，存放整个工程的 main 函数
	Lab4_main.cpp	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.cpp	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.cpp	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.cpp	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.cpp	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.cpp	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
程序自检模块	Lab4_sys.cpp	存放与程序自检功能相关功能函数的函数
	Lab4_sys.h	存放与程序自检功能相关功能函数的函数声明
交互获取文件参数模块	Lab4_fun.cpp	存放与获取文件名与生成条数数目相关功能函数的函数
	Lab4_fun.h	存放与获取文件名与生成条数数目相关功能函数的函数声明

3.5. 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：

```
typedef struct Conf
{
    ……
    int* Array2D;                // 二维数组
    ……
} confvod;
confvod* conf = (confvod*)malloc(sizeof(confvod));
conf->Array2D = (int*)malloc(conf->record_num * 3 * sizeof(int));   // 动态内存分配
```

2. 结构体数组存储方式

结构体数组为 N 行 3 列，动态申请空间，结构体数组声明如下：

```
typedef struct Conf
{
    ……
    DATAITEM* StructArray;   // 结构体数组首指针
    ……
} confvod;
confvod* conf = (confvod*)malloc(sizeof(confvod));
    conf->StructArray = (DATAITEM*)malloc(conf->record_num * sizeof(DATAITEM));// 动态内存分配
```

3. 指针数组存储方式

指针数组为 N 行 3 列，动态申请空间，指针数组声明如下：

```
typedef struct Conf
{
    ……
    DATAITEM** FingerArray; // 指针数组首指针
    ……
} confvod;
confvod* conf = (confvod*)malloc(sizeof(confvod));
```

```
conf->FingerArray = (DATAITEM**)malloc(conf->record_num * sizeof(DATAITEM*)); // 动态内存分配
for (int i = 0; i < conf->record_num; ++i)
{
    conf->FingerArray[i] = (DATAITEM*)malloc(sizeof(DATAITEM));
}
```

4. 链表存储方式

链表为 N 行 3 列，动态申请空间，链表声明如下：

```
typedef struct Conf
{
    ……
    LINKNODE* LinkHead;        // 链表头指针
    ……
} confvod;
confvod* conf = (confvod*)malloc(sizeof(confvod));
LINKNODE* tail = NULL;                    // 初始化（链表）数据加载到链表
for (int i = 0; i <= conf->record_num; ++i)
{
    LINKNODE* newdata;
    newdata = (LINKNODE*)malloc(sizeof(LINKNODE));
    ……  }
```

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
int sys_method	变量	实验 4 工作模式变量，1 表示自动模式，2 表示交互模式
char filepath[MAX_ARRAY_LEN]	变量	交互模式下储存用户输入的文件储存路径
int record_num	变量	交互模式下存储用户输入的数据记录条数
char user_filename[MAX_STR_LEN]	变量	交互界面文件储存路径的临时储存地点
#define MAX_ARRAY_LEN	常量	程序中数组的尺寸常数
#define MAX_STR_LEN	常量	程序中 user_filename 的尺寸常数
double duration	变量	储存排序过程中的时间

3.6. 函数接口设计

整个程序共设计函数 39 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	int argc：命令行参数个数、 char* argv[]:命令行参数值	int（无实际意义）	程序入口函数
run	Lab4_main.cpp	int argc：命令行参数个数、 char* argv[]:命令行参数值	int（无实际意义）	程序主函数，实验 4 函数大体框架
syscheck	Lab4_sys.cpp	无	int； 返回 0 代表程序自检发现错误，结束程序 返回 1 代表程序自检成功，继续程序	程序自检
sysinit	Lab4_sys.cpp	CONF* data 数据参数； confvod* conf 配置文件参数	void	加载系统参数及初始化全局变量
showmainmenu	Lab4_view.cpp	confvod* conf 配置文件参数	int 返回选项	显示输出实验 4 程序的主菜单
showArray2D	Lab4_view.cpp	int n 生成数据条数, confvod* conf 配置文件参数	void	将二维数组显示到屏幕上
showStructArray	Lab4_view.cpp	int n 生成数据条数, confvod* conf 配置文件参数	void	将结构体数组显示到屏幕上
showFingerArray	Lab4_view.cpp	int n 生成数据条数, confvod* conf 配置文件参数	void	将结构体指针数组显示到屏幕上
showLink	Lab4_view.cpp	int n 生成数据条数, confvod* conf 配置文件参数	void	将链表显示到屏幕上
show14menu	Lab4_view.cpp	无	int 返回选项序号	显示功能 14 的修改菜单
callLab3	Lab4_call.cpp	int n 生成数据条数, confvod* conf 配置文件参数	void	调用实验 3 生成数据文件
get_filenum	Lab4_fun.cpp	CONF* data 数据参数； confvod* conf 配置文件参数	int, 如果检查后合法，返回值为 1,否则为 0	通过交互界面获取文件记录条数

续 表 3-3

get_filename	Lab4_fun.cpp	CONF* data 数据参数; confvod* conf 配置文件参数	int, 如果检查后合法，返回值为 1,否则为 0	通过交互界面获取文件储存路径
check_seper_file	Lab4_fun.cpp	char* str:文件路径与文件名 CONF* data 数据参数; confvod* conf 配置文件参数	int, 如果检查后合法，返回值为 1,否则为 0	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
clearstdin	Lab4_fun.cpp	无	int（无实际意义）	对 scanf 输入进行缓存清除
loadToArray2D	Lab4_loadfile.cpp	CONF* data 数据参数; confvod* conf 配置文件参数	int 返回数据记录条数	将数据文件加载到二维数组
loadToStructArray	Lab4_loadfile.cpp	CONF* data 数据参数; confvod* conf 配置文件参数	int 返回数据记录条数	将数据加载到结构体数组
loadToFingerArray	Lab4_loadfile.cpp	CONF* data 数据参数; confvod* conf 配置文件参数	int 返回数据记录条数	将数据加载到结构体指针数组
loadToLink	Lab4_loadfile.cpp	CONF* data 数据参数; confvod* conf 配置文件参数	int 返回数据记录条数	将数据加载到链表
cmpArray2D	Lab4_sort.cpp	const void* a, const void* b	int	二维数组排序比较函数
sortArray2D	Lab4_sort.cpp	int n 生成数据条数, confvod* conf 配置文件参数	void	二维数组排序 qsort 函数
cmpStructArray	Lab4_sort.cpp	const void* a, const void* b	int	结构体数组排序比较函数
sortStructArray	Lab4_sort.cpp	int n 生成数据条数, confvod* conf 配置文件参数	void	结构体数组排序 qsort 函数
cmpFingerArray	Lab4_sort.cpp	const void* a, const void* b	int	结构体指针数组排序排序比较函数
sortFingerArray	Lab4_sort.cpp	int n 生成数据条数, confvod* conf 配置文件参数	void	结构体指针数组排序 qsort 函数

续 表3-3

sortLink1	Lab4_sort.cpp	int n 生成数据条数, confvod* conf 配置文件参数	void	链表排序——冒泡排序
getpar	Lab4_sort.cpp	LINKNODE* begin, LINKNODE* end	LINKNODE*	链表排序——快排比较程序1
sortLink2	Lab4_sort.cpp	LINKNODE* begin, LINKNODE* end	void	链表排序——快排比较程序2
modifyfilesavepath	Lab4_conf.cpp	CONF* data 数据参数; confvod* conf 配置文件参数	void	修改文件存储路径
modifyfilename	Lab4_conf.cpp	CONF* data 数据参数; confvod* conf 配置文件参数	void	修改文件名
modify12	Lab4_conf.cpp	CONF* data 数据参数; confvod* conf 配置文件参数	void	修改第一二个元素的范围
modify3	Lab4_conf.cpp	CONF* data 数据参数; confvod* conf 配置文件参数	void	修改第三个元素的范围
modifynum	Lab4_conf.cpp	CONF* data 数据参数; confvod* conf 配置文件参数	void	修改数据记录条数的范围
modifysysmethod	Lab4_conf.cpp	CONF* data 数据参数; confvod* conf 配置文件参数	void	修改实验4工作模式
freedata	Lab4_conf.cpp	int n 选项 CONF* data 数据参数; confvod* conf 配置文件参数	void	释放动态申请的内存，重新初始化全局变量
free2d	Lab4_conf.cpp	confvod* conf 配置文件参数	void	释放二维数组的内容
freestruct	Lab4_conf.cpp	confvod* conf 配置文件参数	void	释放结构体数组的内容
freeFingerArray	Lab4_conf.cpp	confvod* conf 配置文件参数	void	释放结构体指针数组的内容
freelink	Lab4_conf.cpp	confvod* conf 配置文件参数	void	释放链表的内容

3.7. 函数详细设计

3.7.1. 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。

```
/*********************************************************\
作者信息：
姓名：  张鳃沣   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级张鳃沣个人所有
模块名称：
    实验4 主入口函数
摘要：
    作业，配合实验4
其它说明：
    无。
模块历史：
    张鳃沣于2023年5月5日创建本模块，email: 22281052@bjtu.edu.cn
*********************************************************/
#include "Lab4_HeaderFile.h"

/*
*函数名称：main
*函数功能：实现实验4全部功能,入口函数
*输入参数：int argc：命令行参数个数、char* argv[]:命令行参数值
*返回值：int
*版本信息：create by Lifeng Zhang, 2023-05-05
*/
int main(int argc, char* argv[])
{
    run(argc, argv); // 调用程序主功能实现函数
    return 0;
}
```

图 3-2 程序设计分组训练实验 4 程序入口函数代码截图

3.7.2. 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。

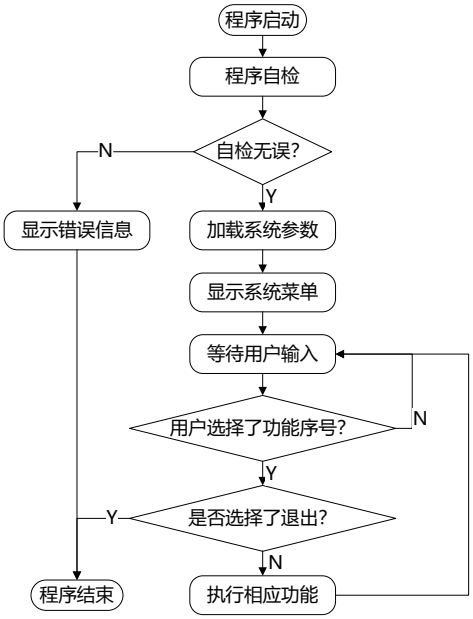

图 3-3 实验 4 主程序流程图

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，argv
  - 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：完成 lab4 的功能实现

3.7.3. 程序自检函数

实验 4 程序自检函数 syscheck 用于进行程序自检功能，函数流程图如图 3-4 所示。

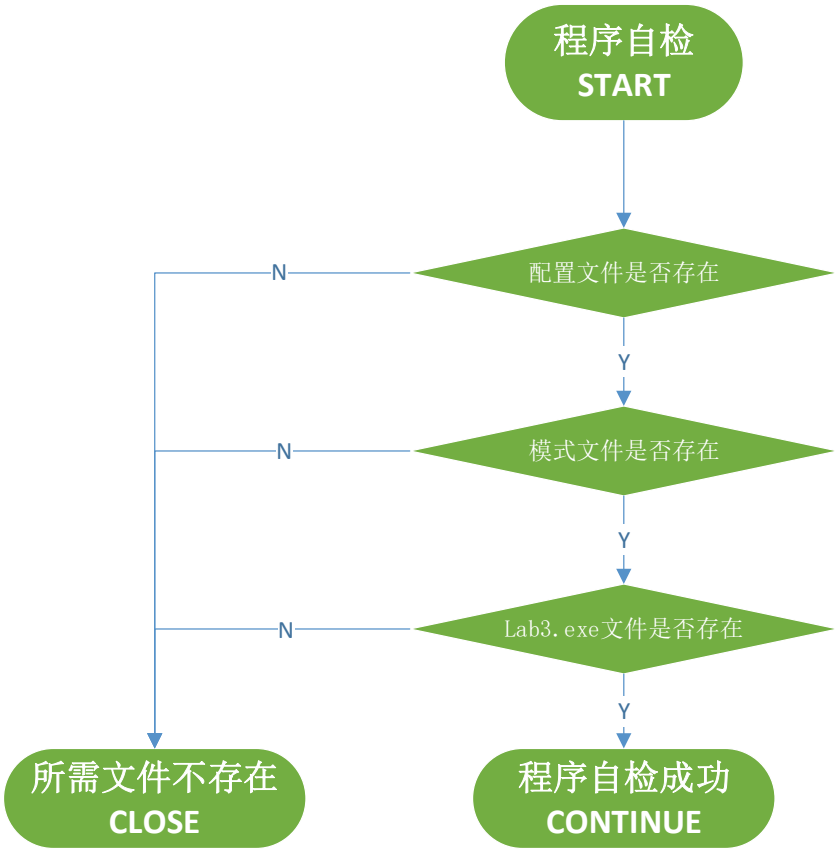

图 3-4 实验 4 程序自检函数流程图

实验 4 程序自检函数 syscheck 的输入参数及函数返回值说明如下：

- 输入参数：无
- 函数返回值：int；
  - 返回 0 代表程序自检发现错误，结束程序
  - 返回 1 代表程序自检成功，继续程序
- 函数功能：进行程序自检

3.7.4. 调用实验 3 生成数据文件函数

实验 4 调用实验 3 生成数据文件函数 syscheck 用于调用实验 3 生成数据文件功能，函数流程图如图 3-5 所示。

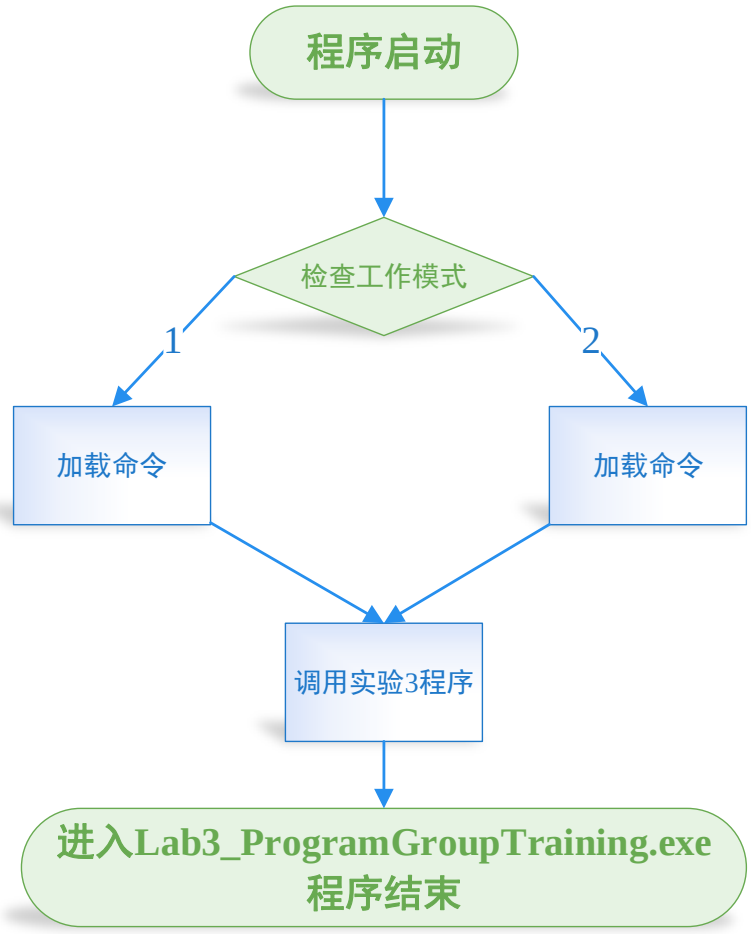

图 3-5 实验 4 调用实验 3 生成数据文件函数流程图

实验 4 程序自检函数 syscheck 的输入参数及函数返回值说明如下：

- 输入参数：无
  - int n 生成数据条数
  - confvod* conf 配置文件参数
- 函数返回值：void
- 函数功能：调用实验 3 生成数据文件

3.7.5. 修改文件名函数

实验 4 修改文件名函数 modifyfilename 用于实现交互模式下修改文件名，函数流程图如图 3-6 所示。

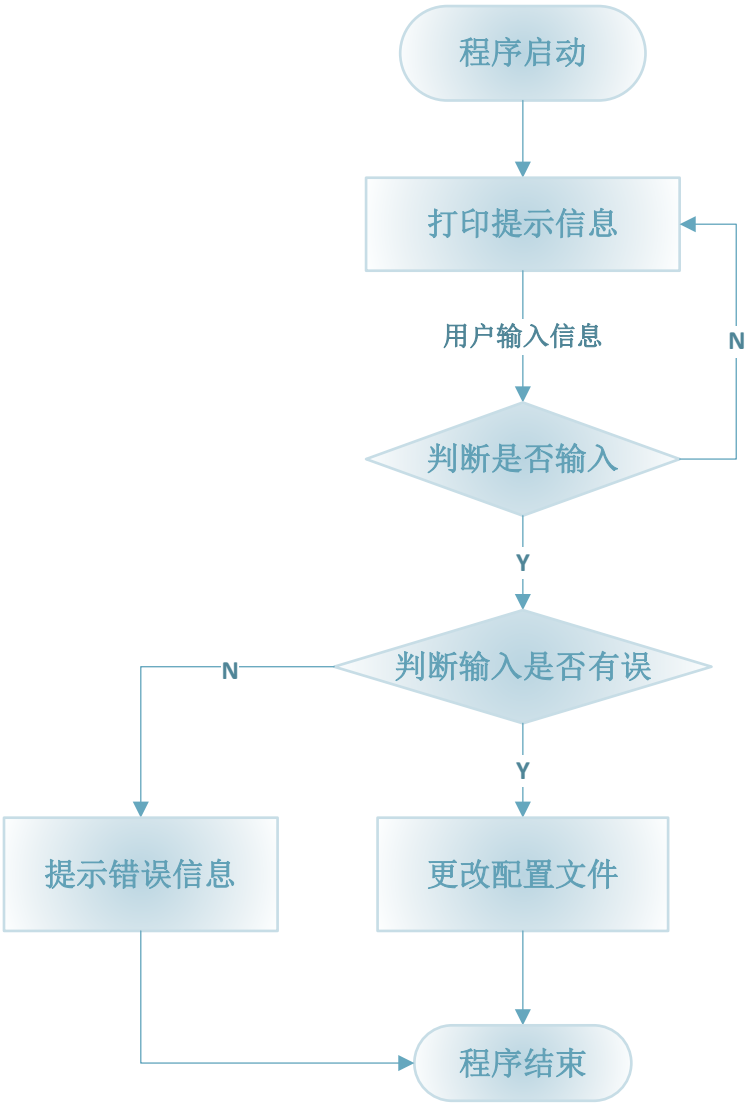

图 3-6 实验 4 修改文件名函数流程图

实验 4 修改文件名函数 modifyfilename 的输入参数及函数返回值说明如下：

- 输入参数：
  - CONF data 数据参数；
  - confvod* conf 配置文件参数
- 函数返回值：void
- 函数功能：修改文件名

3.7.6. 实现内存初始化变量函数

实验 4 实现内存初始化变量函数 freedata 用于实现释放动态申请的内存，重新初始化全局变量功能，函数流程图如图 3-7 所示。

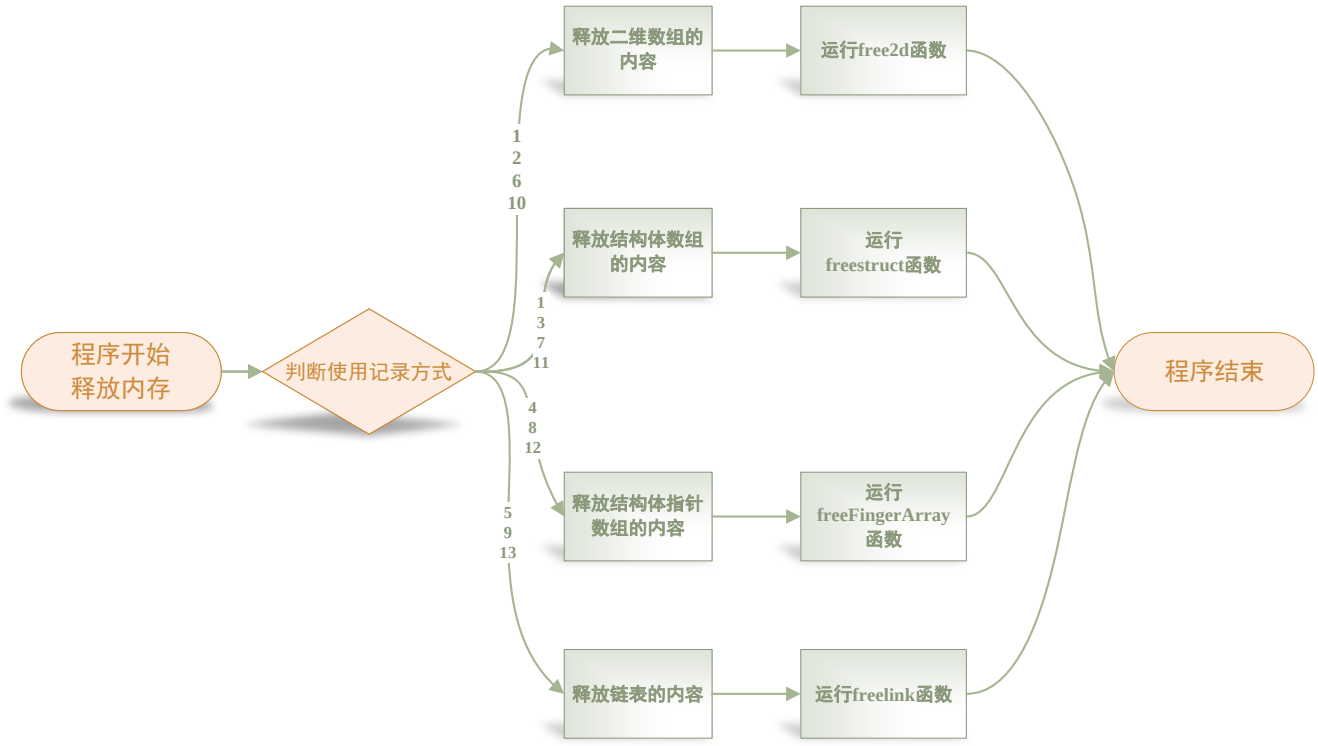

图 3-7 实验 4 内存初始化变量函数 freedata 函数流程图

实验 4 程序内存初始化变量函数 freedata 的输入参数及函数返回值说明如下：

- 输入参数：
  - int n 选项
  - CONF data 数据参数；
  - confvod* conf 配置文件参数
- 函数返回值：void
- 函数功能：释放动态申请的内存，重新初始化全局变量

3.8. 程序交互设计

3.8.1. 系统主菜单设计

系统主菜单显示样式如下：

张鳃丰的实验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:

3.8.2. 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：

(进入修改模式) 请输入想要选择的功能:

1. 修改默认文件存储目录

2. 修改默认文件名

3. 修改第一、二个元素的范围

4. 修改第三个元素的范围

5. 修改数据记录条数的范围

6. 修改实验 4 的工作模式

0. 退出修改模式

3.8.3. 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；

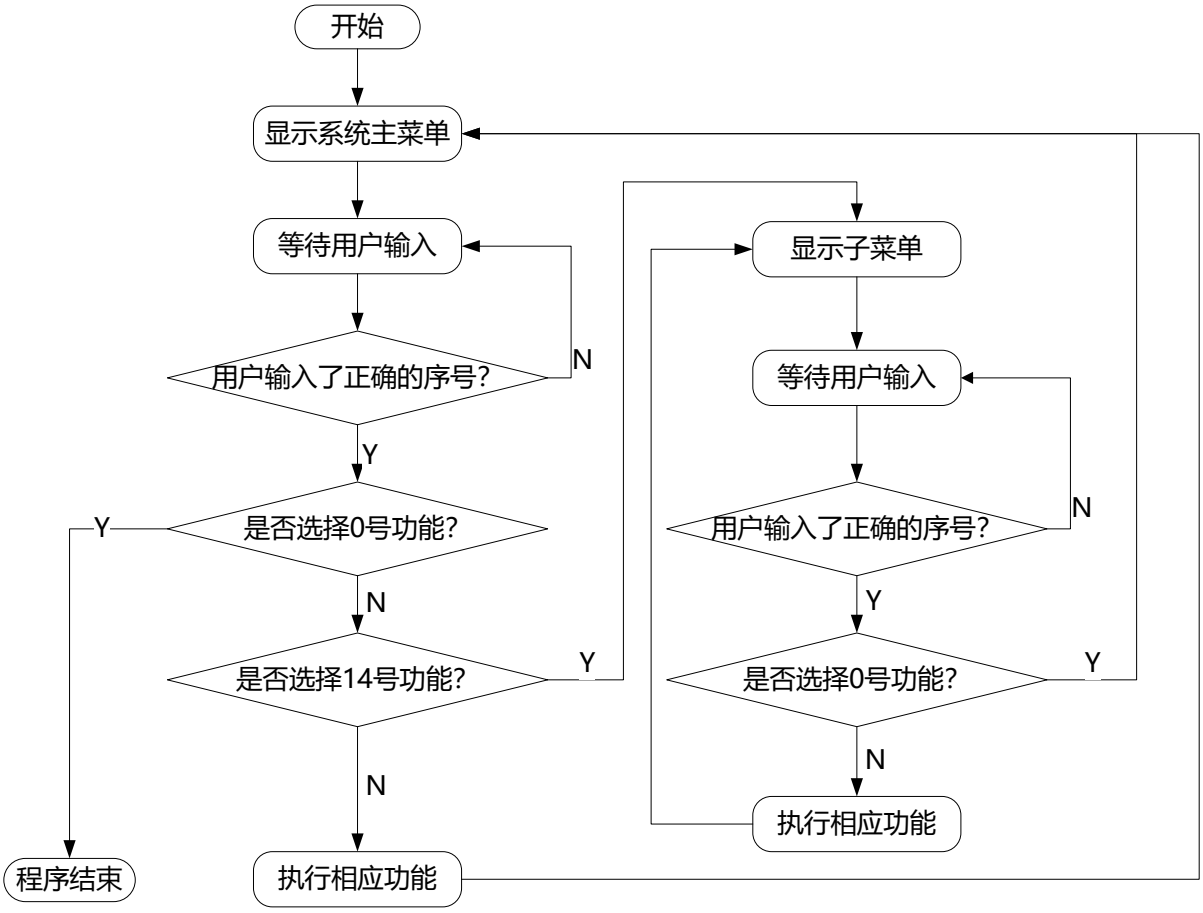

图 3-8 程序菜单显示功能流程图

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;
    }
    }
```

3.8.4. 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：

  请输入正确的文件名及存储路径（可以为绝对路径和相对路径，结尾必须包括.txt）：

- 提示用户输入数据记录条数时的提示信息：

  请输入数据记录条数：

- 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：

  请输入正确的文件存储路径：

- 修改配置文件时提示用户输入新的文件名称的提示信息：

  请输入正确的文件名：

- 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：

  请输入题一二个元素的上限和下限（不输入则使用默认值）

  请输入上限：

  请重新输入上限：

- 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：

  请输入题一二个元素的上限和下限（不输入则使用默认值）

  请输入下限：

  请重新输入下限：

- 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：

  请输入题三个元素的上限和下限（不输入则使用默认值）

  请输入上限：

  请重新输入上限：

- 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：

  请输入题三个元素的上限和下限（不输入则使用默认值）

  请输入下限：

  请重新输入下限：

- 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：

  请输入题记录条数的上限和下限（不输入则使用默认值）

  请输入上限：

  请重新输入上限：

- 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：

  请输入题记录条数的上限和下限（不输入则使用默认值）

  请输入下限：

  请重新输入下限：

- 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：

  请输入实验 4 的工作模式（1 代表自动模式，2 代表交互模式）：

  请重新输入：

3.8.5. 数据记录显示输出交互设计

第一行输出数据记录条数，之后的行依次输出三个数据，用空格分隔。

```
5
14 10 2
5 6 45
6 12 45
18 17 17
19 11 16
```

图 3-9 数据显示样例