

# 北京交通大学 计算机实验报告

## 实验报告（lab2）

年 级：大学一年级  
学 号：20281274  
姓 名：杜海玮  
专 业：计算机科学  
任课老师：邹琪

二零二一年三月

# 目 录

1) 程序实现.....	1
1. 子函数组成.....	1
2. 全局变量组成.....	2
3. 中层函数与 main 函数组成.....	3
2) 实验结果.....	6
1. 第一种情况：用户只启动程序.....	6
2. 第二种情况：用户输入一个字符串.....	7
3. 第三种情况：用户输入两个字符串.....	8
4. 其他情况：用户输入非法输入.....	9
3) 实验总结以及代码的健壮性.....	9
1. 实验总结.....	9
2. 代码的健壮性.....	9
3. 对小组其他成员的文档审查报告.....	10

## 1) 程序实现

### 1. 子函数组成

#### 1.1 字符串转换数字函数

```
void strToNumber(char* c, int *p) { //字符串转换成数字函数
    int res = 0; int j = 0;
    for (int i = strlen(c) - 1; i > -1; i--) {
        res = (c[i] - '0') * ((int)pow(10, j)) + res;
        j++;
    }
    *p = res;
}
```

图 1-1 字符串转换数字函数

输入字符串指针与目标整型指针，利用 strlen 函数读取字符串长度自后往前转换数字。结果输入目标整型指针。

#### 1.2 检查函数

```
int check(char* c) { //用户输入了数据记录条数值或文件名
    int flag = 0;
    for (int i = 0; c[i] != '\0'; i++) {
        if (!isdigit(c[i])) {
            flag = 1; //如果出现非数字字符，则所检查的char数组为文件名
        }
    }
    return flag; //flag为1表示为不只有数字的字符串，为文件名。flag为0表示只有数字的字符串，为文件条数。
}
```

图 1-2 检查函数

输入字符串指针，利用 ctype 库函数中的 isdigit 函数检查是否是纯数字函数，按检查结果返回 1（被检查字符串指针不只有数字，应为文件名）或 0（被检查字符串指针只有数字，应为指定数据条数）。

#### 1.4 生成随机数据函数

```

void CreatRandomData() {
    for (int i = 0; i < Rec; i++) {
        for (int j = 0; j < 3; j++) {
            Data[i][j] = random(100);
        }
    }
}

```

图 1-3 生成随机数据函数

在事先创建的全局二维数组上利用 rand 函数生成随机数据，数据条数由实现创建的数据条数变量 Rec 确定，数据创建后储存进二维数组。

### 1.5 文件写入函数

```

void CreateFile() {
    fopen_s(&fp, FileNameArray, "w");
    if (fp == NULL) {
        printf("file opened failed");
    }
    fprintf(fp, "%d\n", Rec);
    for (int i = 0; i < Rec; i++) {
        int j = 0;
        fprintf(fp, "%d,%d,%d\n", Data[i][j], Data[i][j+1], Data[i][j+2]);
    }

    fclose(fp);
}

```

图 1-5 文件写入函数

将已经存在的文件名 **FileNameArray** 作为目标文件名，打开文件后将之前生成的随机数数组分行录入。如果文件打开失败将提示相关信息。

## 2. 全局变量组成

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#include<ctype.h>
#include<math.h>
#include<string.h>
#include<malloc.h>

#define random(x) (rand()%x)

int Rec;
char* FileNameArray;
FILE* fp;
int Data[11][3];
int* RecArray;
```

图 1-4 全局变量组成

整型 Rec 储存数据条数，字符串指针 FileNameArray 储存文件名地址，文件指针，Data 二维数组储存生成的随机数据，整型指针 RecArray 储存数据条数的地址。

### 3. 中层函数与 main 函数组成

#### 3.1getParam 函数

```
void getParam() { //用户只输入了程序名称用来启动程序
    static char name[20] = { "Document.txt" };
    FileNameArray = name;
    Rec = random(10);
}
```

图 1-5 getParam 函数

该函数时当用户什么都没有输入时启动，将生成默认文件名与随机数据条数（10 条以内）。

### 3.2inputOne 函数

```

void inputOne(char* c) {
    int flag = check(c);
    if (flag == 1) {
        FileNameArray = c;
        Rec = random(10);
    }
    if (flag == 0) {
        RecArray = (int*)malloc(1000 * sizeof(int));
        strToNumber(c, RecArray);
        Rec = *RecArray;
        static char name[20] = { "Document.txt" };
        FileNameArray = name;
    }
}

```

图 1-6 inputOne 函数

该函数在用户仅输入一个输入项时启动，将先调用 check 函数检查输入项是文件名还是数据条数，然后调用相应函数生成随机数据条数或默认文件名，将用户给定的数据条数或文件名放入对应的全局变量中。

### 3.3inputTwo 函数

```

int inputTwo(char* c1, char* c2) {
    if (check(c1) == 1 && check(c2) == 0) {
        FileNameArray = c1;
        RecArray = (int*)malloc(1000 * sizeof(int));
        strToNumber(c2, RecArray);
        Rec = *RecArray;
        return 0;
    }
    if (check(c2) == 1 && check(c1) == 0) {
        FileNameArray = c2;
        RecArray = (int*)malloc(1000 * sizeof(int));
        strToNumber(c1, RecArray);
        Rec = *RecArray;
        return 0;
    }
    else {
        return 1;
    }
}

```

图 1-7 inputTwo 函数

该函数在用户输入两个输入项时启动，将先调用 check 函数检查输入项是文件名还是数据条数，然后将用户给定的数据条数或文件名放入对应的全局变量中。出现情况外的状况将返回错误值 1 终止程序运行。

### 3.3main 函数

```

int main(int argc, char* argv[])
{
    srand((int)time(0)); //设置随机种子

    switch (argc)
    {
        case 1: getParam();
                CreatRandomData();
                CreateFile();
                break; //用户只输入了程序名称用来启动程序

        case 2: inputOne(argv[1]);
                CreatRandomData();
                CreateFile();
                break; //用户输入了数据记录条数值或文件名

        case 3: if (inputTwo(argv[1], argv[2]) == 0) {
                CreatRandomData();
                CreateFile();
                break;
            }
            else {
                printf("wrong input");
                break;
            } //用户输入了数据记录条数值以及文件名

        default: printf("wrong input");
                break;
    }

    return 0;
}

```

图 1-8 main 函数

Main 函数首先设置随机数种子为当前时间 time (0)，然后用 switch-case 结构按照 argv 数值进行分支函数动作。

## 2) 实验结果

### 1. 第一种情况：用户只启动程序





```
Document - 记事本
文件(F) 编辑(E) 格式(O) 查
8
53,5,26
86,27,14
89,38,13
69,0,65
72,12,2
31,81,39
94,41,32
54,73,43
```

图 2-1 用户无输入时默认生成

当用户只启动程序而不给任何输入时，程序调用 `getPara` 子函数，指定文件名为“`Document.txt`”，调用随机数 `random` 函数生成随机数据条数，最后按顺序生成随机数据，写入文件。

## 2. 第二种情况：用户输入一个字符串

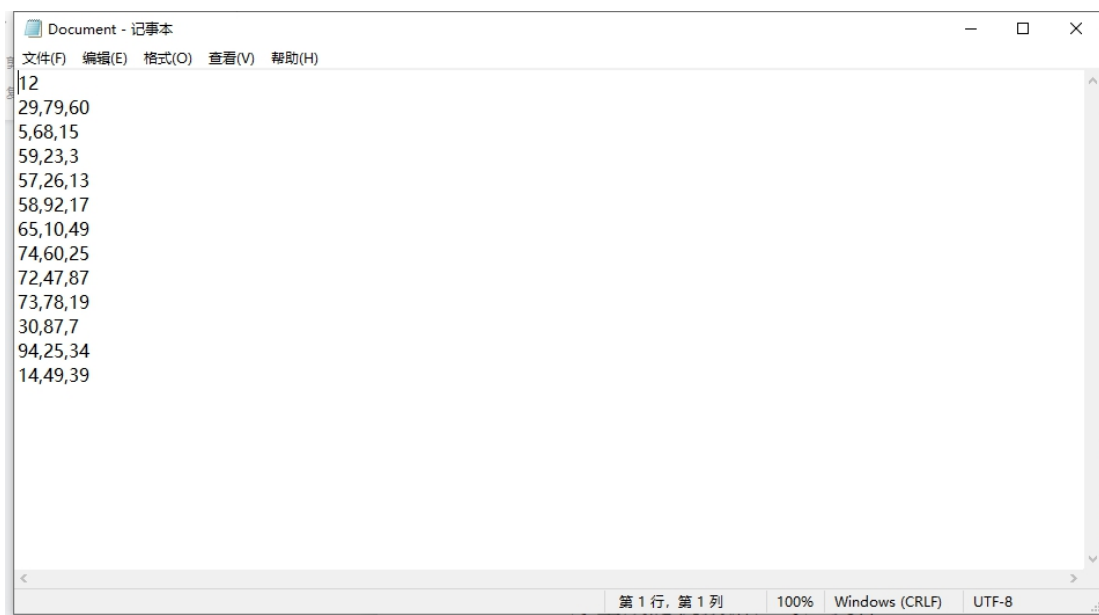


图 2-2 用户输入一个纯数字字符串（12）时生成

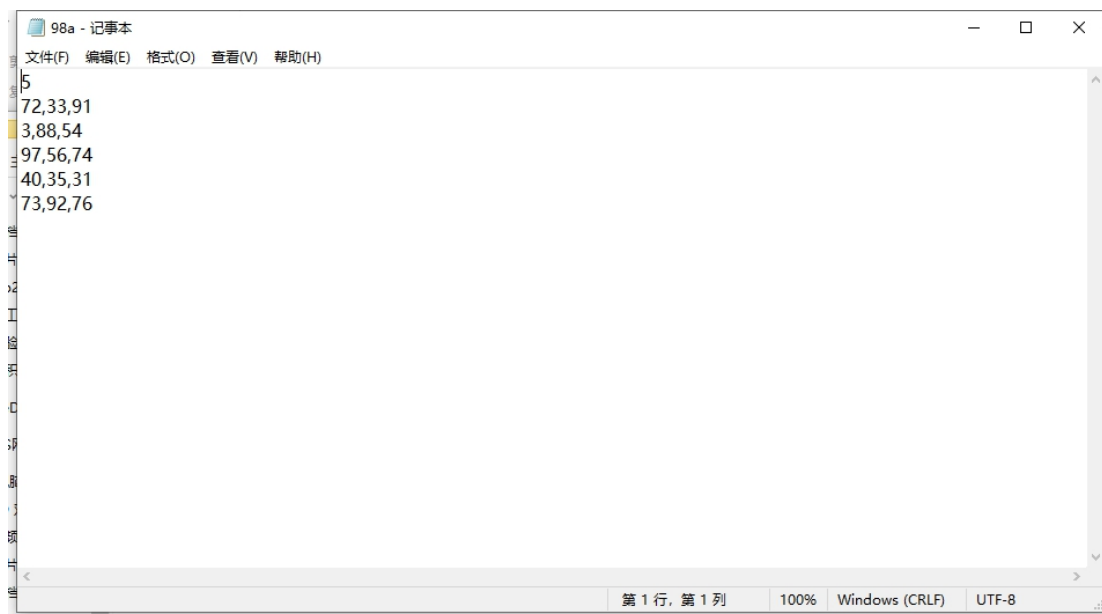


图 2-3 用户输入一个文件名字符串（98a）时生成

当用户输入一个字符串时，程序调用 inputOne 函数，检验其为纯数字字符串或文件名字符串（非纯数字），然后根据结果调用同上函数生成指定数据量或指定文件名的随机数据文件。

### 3. 第三种情况：用户输入两个字符串



图 2-4 用户同时指定两个参数（2abddfa213,6）的情况

当用户输入两个字符串时，程序调用 inputTwo 函数，依次检验其为纯数字字符串或文件名字符串（非纯数字），然后根据结果调用同上函数生成指定数据

量与文件名的随机数据文件。针对同时出现输入一个类型字符串的情况，将显示 `wrong input` 与输入提示并跳出循环。

#### 4. 其他情况：用户输入非法输入

针对以上三种情况之外的情况，程序调用 `default` 选项，显示 `wrong input` 与输入提示并跳出。

### 3) 实验总结以及代码的健壮性

#### 1. 实验总结

通过本实验，我深刻认识了模块化程序的模式，通过事先声明与引用函数，`main` 函数里只需要提纲挈领地放置若干中层函数，简洁美观地实现了既定功能。

在实验中我遇到的主要问题是如何进行输入鉴定，最后采用了返回值与逻辑判断结构地方法，用比较易读地方式实现了该功能。正如 2. 所言，通过阅读关于代码健壮性地资料，我发现我的代码还没有剔除极端数据（大于 65535 的数）的判定。

#### 2. 代码的健壮性

##### 2.1 定义

健壮性是指软件（代码）对于规范要求以外的输入情况的处理能力。

##### 2.2 结合健壮性的概念谈谈程序还有什么地方可以改进

答：程序在用户引导方面可以“手把手”地教用户如何输入，在我的程序中，出现错误时除了 `wrong input` 外还会显示输入提示以便用户正确调用函数。在输入方面，鉴于字符串转换数字时的固有局限（例如用户输入 013123568978，或者 -4654654654 这些都是无法转换的），我的程序还可以增加判定环节，将这些无效输入剔除。

### 3. 对小组其他成员的文档审查报告

陈明强同学：他的文档用图多且详实，但代码方面可能有些复杂了。

高原同学：他的实验报告用图多，而且在需要放大的地方用了放大到具体代码的截图，值得学习。