

实验二 实验报告

王子龙

CIT1808

18281218

Mar. 12th, 2019

目录

一、程序设计	- 1 -
(一)、函数设计	- 1 -
1. <i>getTxt(char str[])</i>	- 1 -
2. <i>getNumofStr(char str[])</i>	- 1 -
3. <i>writeArray(int dArray[][3], int n)</i>	- 2 -
4. <i>getCmd(int *Num,char Name[],int argc,char *argv[])</i>	- 2 -
(二)、MAIN 代码逻辑	- 4 -
1. <i>main</i> 函数代码	- 4 -
2. 变量	- 5 -
3. 程序逻辑	- 5 -
(三)、程序运行效果	- 5 -
1. 命令行参数为空	- 5 -
2. 只输入文件名	- 6 -
3. 只输入元素组数	- 6 -
4. 先输入文件名再输入元素组数	- 6 -
5. 先输入元素组数再输入文件名	- 7 -
(四)、程序设计的缺陷与遇到的问题	- 7 -
二、程序健壮性	- 7 -
(一)、程序健壮性的概念	- 7 -
(二)、关于前文代码的健壮性	- 8 -
附件 1：完整代码	- 1 -

一、程序设计

(一)、函数设计

1. getTxt(char str[])

函数代码：

```
int getTxt(char str[]) {  
    if(strstr(str, ".txt")) return 1;  
    else if(strstr(str, ".Txt")) return 1;  
    else if(strstr(str, ".tXt")) return 1;  
    else if(strstr(str, ".TXt")) return 1;  
    else if(strstr(str, ".txT")) return 1;  
    else if(strstr(str, ".TxT")) return 1;  
    else if(strstr(str, ".TXT")) return 1;  
    else if(strstr(str, ".tXT")) return 1;  
    else return 0;  
}
```

代码解释：

通过使用 string.h 里面的 strstr (const char * str1, const char * str2)函数判断字符串内是否含有字符 “.txt”。

传入参数：str[]

返回值(int)：

0	传入的字符串中含有 “.txt”
1	传入的字符串中不含有 “.txt”

表 1-1

2. getNumofStr(char str[])

函数代码：

```
int getNumofStr(char str[]) {  
    int i = 0;  
    while(str[i] != '\0') {  
        i++;  
    }  
    return i;  
}
```

代码解释：判断一个字符串的长度（不包括' \0' ）。

传入参数：str[]

返回值(int)：字符串长度

3. writeArray(int dArray[][3], int n)

函数代码：

```
void writeArray(int dArray[][3], int n) {
    for (int i = 0; i < n; ++i) {
        dArray[i][0] = rand() % 1000;
        dArray[i][1] = rand() % 1000;
        dArray[i][2] = rand() % 1000;
    }
}
```

代码解释：将一个 n 行 3 列的 int 型二维数组内的每个元素写成一个 3 位随机数。

传入参数：

dArray[][3]	列数为 3 的二维数组
n	数组行数

表 1-2

4. getCmd(int *Num,char Name[],int argc,char *argv[])

函数代码：

```
int getCmd(int *Num,char Name[],int argc,char *argv[]) {
    if (argc == 1) return 0;
    else if (argc == 2) {
        if (getTxt(argv[1])){
            int i = 0;
            while(argv[1][i] != '\0') {
                Name[i] = argv[1][i];
                i++;
            }
            Name[i+1] = '\0';
            return 1;
        }
    }
    else {
        for (int i = 0; i < getNumofStr(argv[1]); ++i) {
            *Num += (argv[1][i] - 48) * pow(10,getNumofStr(argv[1]) - i - 1);
        }
        return 2;
    }
}
```

```

    }
}
else if (argc == 3) {
    if (getTxt(argv[1])) {
        int i = 0;
        while(argv[1][i] != '\0') {
            Name[i] = argv[1][i];
            i++;
        }
        Name[i+1] = '\0';
        for (int i = 0; i < getNumofStr(argv[2]); ++i) {
            *Num += (argv[2][i] - 48) * pow(10,getNumofStr(argv[2]) - i - 1);
        }
    }
    else {
        int i = 0;
        while(argv[2][i] != '\0') {
            Name[i] = argv[2][i];
            i++;
        }
        Name[i+1] = '\0';
        for (int i = 0; i < getNumofStr(argv[1]); ++i) {
            *Num += (argv[1][i] - 48) * pow(10,getNumofStr(argv[1]) - i - 1);
        }
    }
    return 3;
}
else return 4;
}

```

代码解释：

判断传入值的数量以及文件名和数据组数之间的顺序关系

传入参数：

*Num	用于获取 main 函数之中存放元素组数的变量(int)地址
Name[]	用于获取 main 函数之中存放文件名的字符串
argc	用于获取 main 函数之中的 argc
*argv[]	用于获取 main 函数之中的 argv

表 1-3

返回值(int)：

0	在命令行参数中既没有输入文件名，又没有输入元素组数
1	在命令行参数中只输入了文件名
2	在命令行参数中只输入了元素组数
3	在命令行参数中既输入了文件名，又输入了元素组数
4	其他情况

表 1-4

(二)、main 代码逻辑

1. main 函数代码

```
int main(int argc, char *argv[]) {
    int num = 0;
    char name[10000];
    srand((unsigned)time(0));
    switch (getCmd(&num, name, argc, argv)) {
        case 0: {
            strcpy(name, "RandomValue.txt");
            num = rand() % 1000;
            break;
        }
        case 1: {
            num = rand() % 1000;
            break;
        }
        case 2: {
            strcpy(name, "RandomValue.txt");
            break;
        }
        case 3:
            break;
        case 4:
            printf("Error!\n");
            break;
        default:
            printf("Error!\n");
            break;
    }

    int a[num][3];
    writeArray(a, num);
    FILE *fp = fopen(name, "w+");
    fprintf(fp, "%d\n", num);
    for (int i = 0; i < num; ++i) {
        fprintf(fp, "%3d,%3d,%3d\n", a[i][0], a[i][1], a[i][2]);
    }
    fclose(fp);

    return 0;
}
```

2. 变量

变量名	变量类型	变量作用
num	int	用于储存元素组数
name	char[]	用于储存文件名
a[num][3]	int[][]	用于储存各个元素的值

表 2-1

3. 程序逻辑

首先声明变量并为 `ran()`函数通过时间取得种子，然后通过 `getCmd` 函数来判断在命令行之中用户所输入的文件名以及数据组数的状态，并根据要求，对未输入的文件名和数据组数进行处理，随后通过 `writeArray` 函数将各个元素写上随机数，并生成文件，将数据写入其中，关闭文件。

(三)、程序运行效果

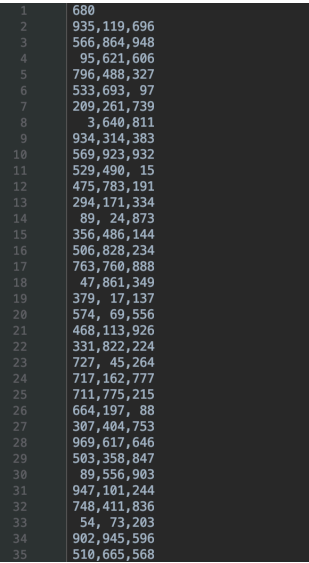
1. 命令行参数为空

命令行：

```
[(base) wangzilongdeMacBook-Pro:Lab2_18281218 wangzilong$ ./a.out
```

图 3 - 1

文件效果



```
1 680
2 935,119,696
3 566,864,948
4 95,621,606
5 796,488,327
6 533,693, 97
7 209,261,739
8 3,640,811
9 934,314,383
10 569,923,932
11 529,490, 15
12 475,783,191
13 294,171,334
14 89, 24,873
15 356,486,144
16 506,828,234
17 763,760,888
18 47,861,349
19 379, 17,137
20 574, 69,556
21 468,113,926
22 331,822,224
23 727, 45,264
24 717,162,777
25 711,775,215
26 664,197, 88
27 307,404,753
28 969,617,646
29 503,358,847
30 89,556,903
31 947,101,244
32 748,411,836
33 54, 73,203
34 902,945,596
35 510,665,568
```

图 3-2

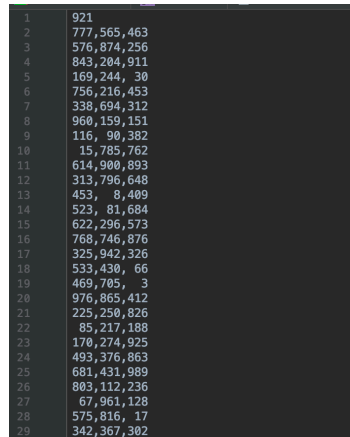
2. 只输入文件名

命令行：

```
(base) wangzilongdeMacBook-Pro:Lab2_18281218 wangzilong$ ./a.out a.txt
```

图 3-3

a.txt:



1	921
2	777,565,463
3	576,874,256
4	843,204,911
5	169,244,30
6	756,216,453
7	338,694,312
8	960,159,151
9	116,90,382
10	15,785,762
11	614,900,893
12	313,796,648
13	453,8,409
14	523,81,694
15	622,296,573
16	768,746,876
17	325,942,326
18	533,430,66
19	469,705,3
20	976,865,412
21	225,250,826
22	85,217,188
23	170,274,925
24	493,376,863
25	681,431,989
26	803,112,236
27	67,961,128
28	575,816,17
29	342,367,302

图 3-4

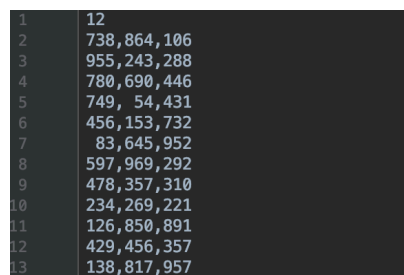
3. 只输入元素组数

命令行：

```
(base) wangzilongdeMacBook-Pro:Lab2_18281218 wangzilong$ ./a.out 12
```

图 3-5

RandomValue.txt:



1	12
2	738,864,106
3	955,243,288
4	780,690,446
5	749,54,431
6	456,153,732
7	83,645,952
8	597,969,292
9	478,357,310
10	234,269,221
11	126,850,891
12	429,456,357
13	138,817,957

图 3-6

4. 先输入文件名再输入元素组数

命令行：

```
(base) wangzilongdeMacBook-Pro:Lab2_18281218 wangzilong$ ./a.out b.txt 3
```

图 3-7

b.txt :

1	3
2	16,623,296
3	997,409,178
4	380,614,653

图 3-8

5. 先输入元素组数再输入文件名

命令行 :

```
(base) wangzilongdeMacBook-Pro:Lab2_18281218 wangzilong$ ./a.out 5 c.txt
```

图 3-9

c.txt :

8	1	5
	2	558,606,274
	3	42,110,129
	4	809, 98,907
	5	868,754,986
	6	624,250,294

图 3-10

(四)、程序设计的缺陷与遇到的问题

本程序设计存在着一定的缺陷和问题, 首先, 代码中区分文件名和元素组数的方法是判断哪个字符串中含有“.txt” 或是否含有“.txt”, 但是当输入的文件名当中不含有“.txt” 时, 程序的运行将出现较大的问题, 因此该代码运行的基础是, 在命令行参数输入的过程之中, 要出现“.txt” 片段, 此外, 函数中还有部分运算时可以通过 c 语言的库函数实现的, 例如判断一个字符串的长度, 将字符串转换成 int 等。

二、程序健壮性

(一)、程序健壮性的概念

健壮性是指软件对于规范要求以外的输入情况的处理能力。

所谓健壮的系统是指对于规范要求以外的输入能够判断出这个输入不符合规范要求, 并能合理的处理方式。

另外健壮性有时也和容错性, 可移植性, 正确性有交叉的地方。

比如, 一个软件可以从错误的输入推断出正确合理的输入, 这属于容错性量度标准, 但是也可以认为这个软件是健壮的。

一个软件可以正确地运行在不同环境下, 则认为软件可移植性高, 也可以叫, 软件在不

同平台下是健壮的。

一个软件能够检测自己内部的设计或者编码错误，并得到正确的执行结果，这是软件的正确性标准，但是也可以说，软件有内部的保护机制，是模块级健壮的。

软件健壮性是一个比较模糊的概念，但是却是非常重要的软件外部量度标准。软件设计的健壮与否直接反应了分析设计和编码人员的水平。

(二)、关于前文代码的健壮性

前文代码的健壮性并不是很好，因为前文代码之中有很多情况并没有考虑周全，导致如果以该种方式运行，程序会出现错误甚至崩溃，如：文件名不含有“.txt”时，命令行输入了超过两组数据是，数字含有字母中文时，数据过大时。

改进方法，增加函数分支，对特殊情况以及非法情况进行判断并给出相应的处理结果。如：将判断“.txt”改为判断数字，效果就会好不少。

附件 1：完整代码

```
/******|
作者信息:
    姓名: 王子龙
版权声明:
    版权由王子龙所有
模块名称:
    实验2 (输出随机数)
摘要:
    文件、命令行参数、随机数生成、数组、指针、结构的学习
其它说明:

模块历史:
    王子龙 2019 年3 月12 日创建该文件 wangzilong0201@gmail.com

*****/
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <time.h>

int getCmd(int *Num,char Name[],int argc,char *argv[]);
int getTxt(char str[]);
int getNumofStr(char str[]);
void writeArray(int dArray[][3], int n);

int main(int argc,char *argv[]) {
    int num = 0;
    char name[10000];
    srand((unsigned)time(0));

    switch (getCmd(&num,name,argc,argv)) {
        case 0:{
            strcpy(name,"RandomValue.txt");
            num = rand() % 1000;
            break;
        }
        case 1:{
            num = rand() % 1000;
            break;
        }
    }
}
```

```

    }
    case 2:{
        strcpy(name,"RandomValue.txt");
        break;
    }
    case 3:
        break;
    case 4:
        printf("Error!\n");
        break;
    default:
        printf("Error!\n");
        break;
}

int a[num][3];

writeArray(a,num);

FILE *fp = fopen(name,"w+");

fprintf(fp,"%d\n",num);
for (int i = 0; i < num; ++i) {
    fprintf(fp,"%3d,%3d,%3d\n",a[i][0],a[i][1],a[i][2]);
}

fclose(fp);

return 0;
}

/*****
* 函数名称: getCmd
* 函数作用: 获取在命令行参数传入的数据组数和文件名
* 输入参数: int *Num,char Name[],int argc,char *argv[]
* 返回值: 在命令行输入的值的状态
*****/

int getCmd(int *Num,char Name[],int argc,char *argv[]) {
    if (argc == 1) return 0;
    else if (argc == 2) {
        if (getTxt(argv[1])){
            int i = 0;
            while(argv[1][i] != '\0') {

```

```

        Name[i] = argv[1][i];
        i++;
    }
    Name[i+1] = '\0';
    return 1;
}

else {
    for (int i = 0; i < getNumofStr(argv[1]); ++i) {
        *Num += (argv[1][i] - 48) * pow(10,getNumofStr(argv[1]) - i - 1);
    }
    return 2;
}
}

else if (argc == 3) {
    if (getTxt(argv[1])) {
        int i = 0;
        while(argv[1][i] != '\0') {
            Name[i] = argv[1][i];
            i++;
        }
        Name[i+1] = '\0';
        for (int i = 0; i < getNumofStr(argv[2]); ++i) {
            *Num += (argv[2][i] - 48) * pow(10,getNumofStr(argv[2]) - i - 1);
        }
    }
    else {
        int i = 0;
        while(argv[2][i] != '\0') {
            Name[i] = argv[2][i];
            i++;
        }
        Name[i+1] = '\0';
        for (int i = 0; i < getNumofStr(argv[1]); ++i) {
            *Num += (argv[1][i] - 48) * pow(10,getNumofStr(argv[1]) - i - 1);
        }
    }
    return 3;
}

else return 4;
}
}

```

/******

* 函数名称: getTxt

* 函数作用: 判断字符串内是否含有".txt"

```

* 输入参数: str[]
* 返回值: 是或否
*****/

int getTxt(char str[]) {
    if(strstr(str,".txt")) return 1;
    else if(strstr(str,".Txt")) return 1;
    else if(strstr(str,".tXt")) return 1;
    else if(strstr(str,".TXt")) return 1;
    else if(strstr(str,".txT")) return 1;
    else if(strstr(str,".TxT")) return 1;
    else if(strstr(str,".TXT")) return 1;
    else if(strstr(str,".tXT")) return 1;
    else return 0;
}

/*****
* 函数名称: getNumofStr
* 函数作用: 获取字符串长度
* 输入参数: str[]
* 返回值: 字符串长度
*****/

int getNumofStr(char str[]) {
    int i = 0;
    while(str[i] != '\0') {
        i++;
    }
    return i;
}

/*****
* 函数名称: writeArray
* 函数作用: 将随机数写入数组
* 输入参数: *dArray[],n
* 返回值: N/A
*****/

void writeArray(int dArray[][3], int n) {
    for (int i = 0; i < n; ++i) {
        dArray[i][0] = rand() % 1000;
        dArray[i][1] = rand() % 1000;
        dArray[i][2] = rand() % 1000;
    }
}

```

}
}