个人实践作业二：代码规范及代码复审

**本人学号：202131060719**

**本人姓名：刘袁晨昕**

**复审对象学号：202131061320**

**复审对象GitHub远程仓库地址：**[**https://github.com/zhangsan8296/WordCount.git**](https://github.com/zhangsan8296/WordCount.git)

**复审对象源程序代码：**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_FILE\_NAME\_LENGTH 256

#define MAX\_WORD\_LENGTH 256

int count\_char(FILE \*fp)

{

int count = 0;

char c;

while ((c = fgetc(fp)) != EOF)

{

count++;

}

return count;

}

int count\_word(FILE \*fp)

{

int count = 0;

char word[MAX\_WORD\_LENGTH];

int i, len;

while (fscanf(fp, "%s", word) != EOF)

{

len = strlen(word);

for (i = 0; i < len; i++)

{

if (word[i] != ',' && word[i] != ' ')

{

count++;

// skip the rest of the word

while (i < len && word[i] != ',' && word[i] != ' ')

{

i++;

}

}

}

}

return count;

}

int main(int argc, char \*argv[])

{

if (argc < 2 || argc > 3)

{

printf("Usage: %s [-c | -w] [input\_file\_name]\n", argv[0]);

return 1;

}

char \*mode = argv[1];

if (strcmp(mode, "-c") != 0 && strcmp(mode, "-w") != 0)

{

printf("Error: invalid parameter\n");

return 1;

}

char file\_name[MAX\_FILE\_NAME\_LENGTH] = "";

if (argc == 3)

{

strncpy(file\_name, argv[2], MAX\_FILE\_NAME\_LENGTH);

}

else

{

printf("Please input file name: ");

scanf("%s", file\_name);

}

FILE \*fp = fopen(file\_name, "r");

if (fp == NULL)

{

printf("Error: cannot open file %s\n", file\_name);

return 1;

}

int count;

if (strcmp(mode, "-c") == 0)

{

count = count\_char(fp);

printf("字符数%d\n", count);

}

else

{

count = count\_word(fp);

printf("单词数%d\n", count);

}

fclose(fp);

return 0;

}

**阅读理解代码：**

代码中使用了stdio.h、stdlib.h和string.h三个头文件，分别用于输入输出、内存分配和字符串操作。同时，定义了两个常量：MAX\_FILE\_NAME\_LENGTH和MAX\_WORD\_LENGTH，分别表示文件名和单词的最大长度。

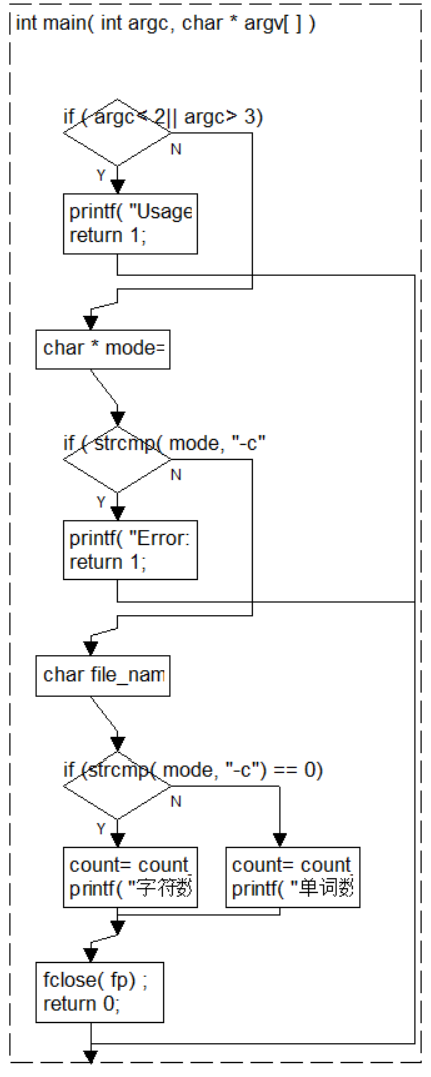
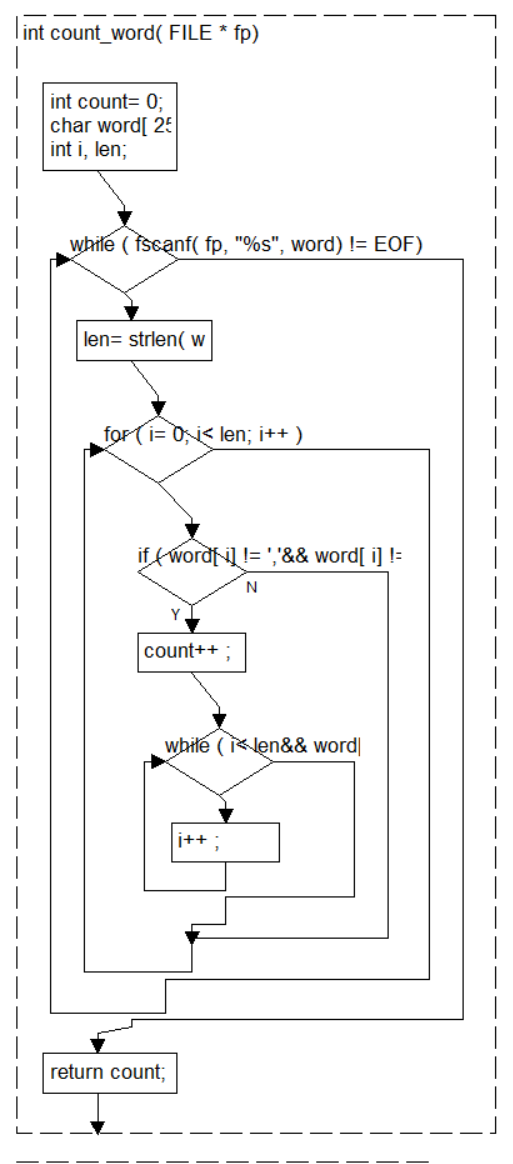
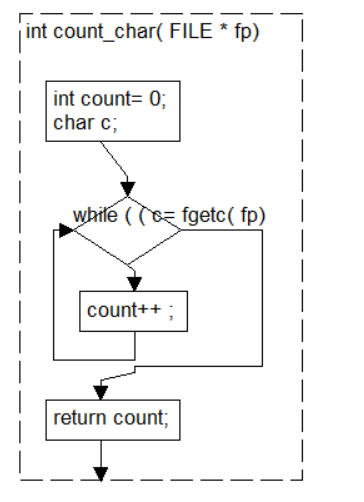
1. 定义两个函数：count\_char和count\_word，分别用于计算字符数和单词数

count\_char函数使用fgetc函数逐个读取文件中的字符，直到读到文件结尾EOF，统计字符数并返回。count\_word函数使用fscanf函数逐个读取文件中的单词，跳过单词中的空格和逗号，统计单词数并返回。

3.获取命令行参数中的模式和文件名，如果没有指定文件名则从标准输入中获取

使用strcmp函数比较模式参数是否为"-c"或"-w"，如果不是则输出错误信息并退出程序。如果指定了文件名，则使用strncpy函数将文件名复制到file\_name数组中，否则从标准输入中获取文件名。

**代码流程图：**



**复审结果（改进建议）：**

1. 在count\_word函数中，使用了fscanf函数读取单词，但是这种方法可能会导致缓冲区溢出的问题。建议使用fgets函数读取一行文本，然后使用strtok函数分割单词。
2. 在count\_word函数中，使用了一个嵌套的循环来跳过单词中的空格和逗号，这种方法比较慢。建议使用一个指针来遍历单词，这样可以避免重复遍历单词。
3. 在count\_word函数中，没有对文件读取失败的情况进行处理，可能会导致程序崩溃。建议在读取单词时，检查文件是否读取成功，如果读取失败则输出错误信息并退出程序。

**总结：**

上述代码是一个简单的统计文件字符数和单词数的程序，主要分为三个部分：定义两个函数、main函数和一些常量和头文件的引入。其中，count\_char函数用于计算文件中的字符数，count\_word函数用于计算文件中的单词数，main函数用于获取命令行参数、打开文件、调用相应的函数计算字符数或单词数，并输出结果。

在代码实现中，使用了一些C语言的基本语法和函数，如fopen、fgetc、fscanf、printf等。同时，代码中也存在一些问题，如缓冲区溢出、文件读取失败等，需要进行改进。