实验二

17281219 孙嘉葳

指导老师：田媚

目录

[1. 按如下函数原型要求编写函数，统计给定一行英文字符串（字符串首地址放在参数szLine）中的英文单词个数（假设所有以分隔符分隔的字符串都是一个词）。编写测试函数对该函数功能进行测试验证。 3](#_Toc509833776)

[2. 1）请认真参看文档：实验 2 结构说明.pptx 和CordeForLab2.h 中的类型说明。本实验重要考查点： 3](#_Toc509833777)

[2）阅读函数 AllocateSpaceForTextFilesInfo，请解释英文单词 allocate 和 memory 的含义。查阅资料并在实验报告中说明函数 malloc的具体功能。 4](#_Toc509833778)

[3）阅读函数 CountWordsOfEuropeanTxtFile，画出该函数流程图，查阅资料并在实验报告中说明其中的函数fopen或fclose的具体功能。 5](#_Toc509833779)

[4）阅读CodeForLab2.cpp的各个函数的功能说明，补全代码实现如下各个函数的功能： 6](#_Toc509833780)

[5）调用 TestCountFilesInfo验证各功能正确性。 6](#_Toc509833781)

[6）阅读函数 ExportFilesInfoToWebPage，查阅资料并在实验报告中说明函数fprintf 的功能。 6](#_Toc509833782)

[6](#_Toc509833783)

[3.思考： 7](#_Toc509833784)

[一． Testcountfilesinfo()函数： 7](#_Toc509833785)

[二． ALinkMaker()函数 9](#_Toc509833786)

1. 按如下函数原型要求编写函数，统计给定一行英文字符串（字符串首地址放在参数szLine）中的英文单词个数（假设所有以分隔符分隔的字符串都是一个词）。编写测试函数对该函数功能进行测试验证。

如所提交的源代码。

1. 1）请认真参看文档：实验 2 结构说明.pptx 和CordeForLab2.h 中的类型说明。本实验重要考查点：
2. 是否能完全理解 CodeForLab2.h 中的两个结构体类型 TEXTFILEINFO 和 FILESINFO 及其间关系。

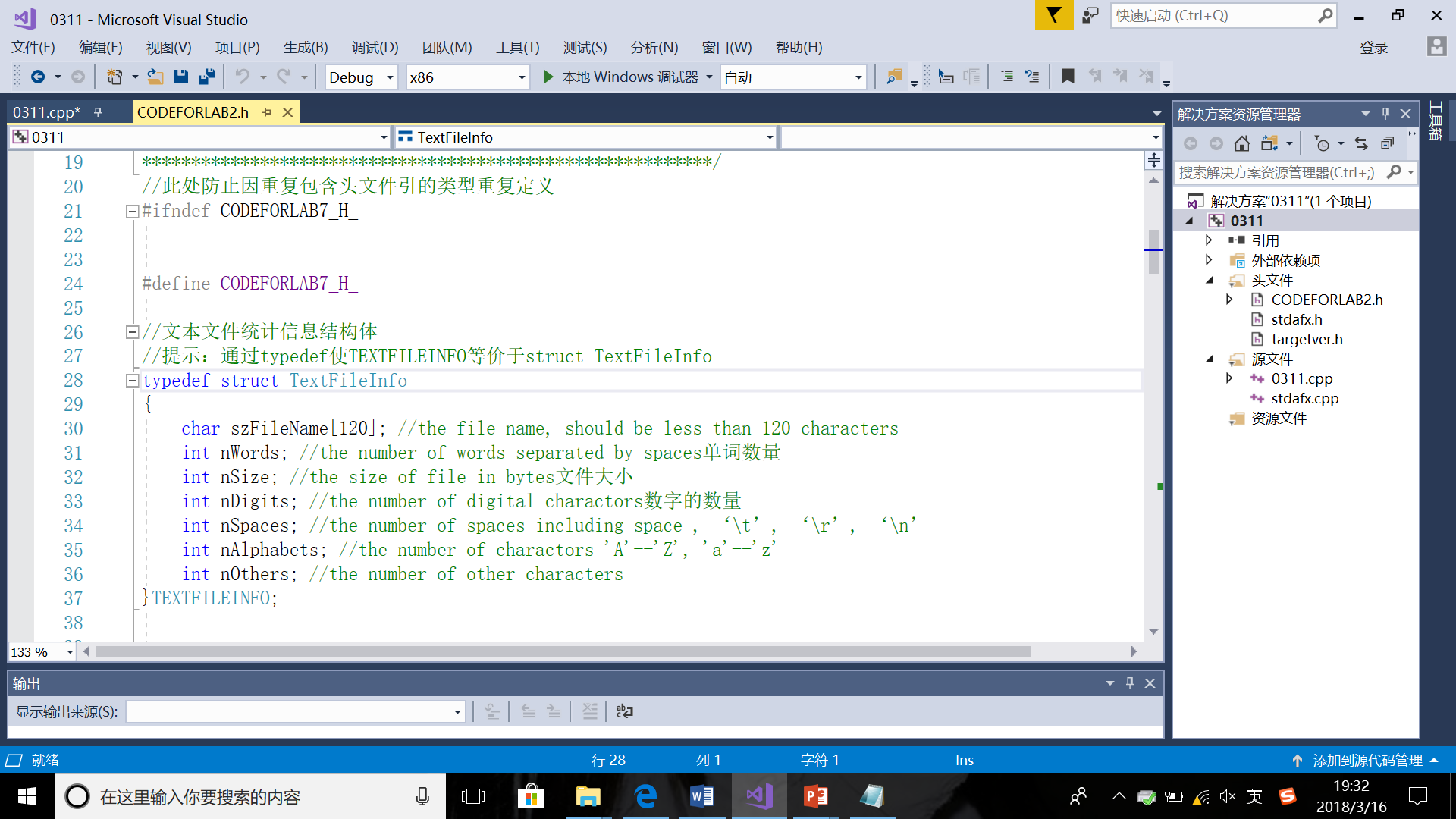


图 2.1

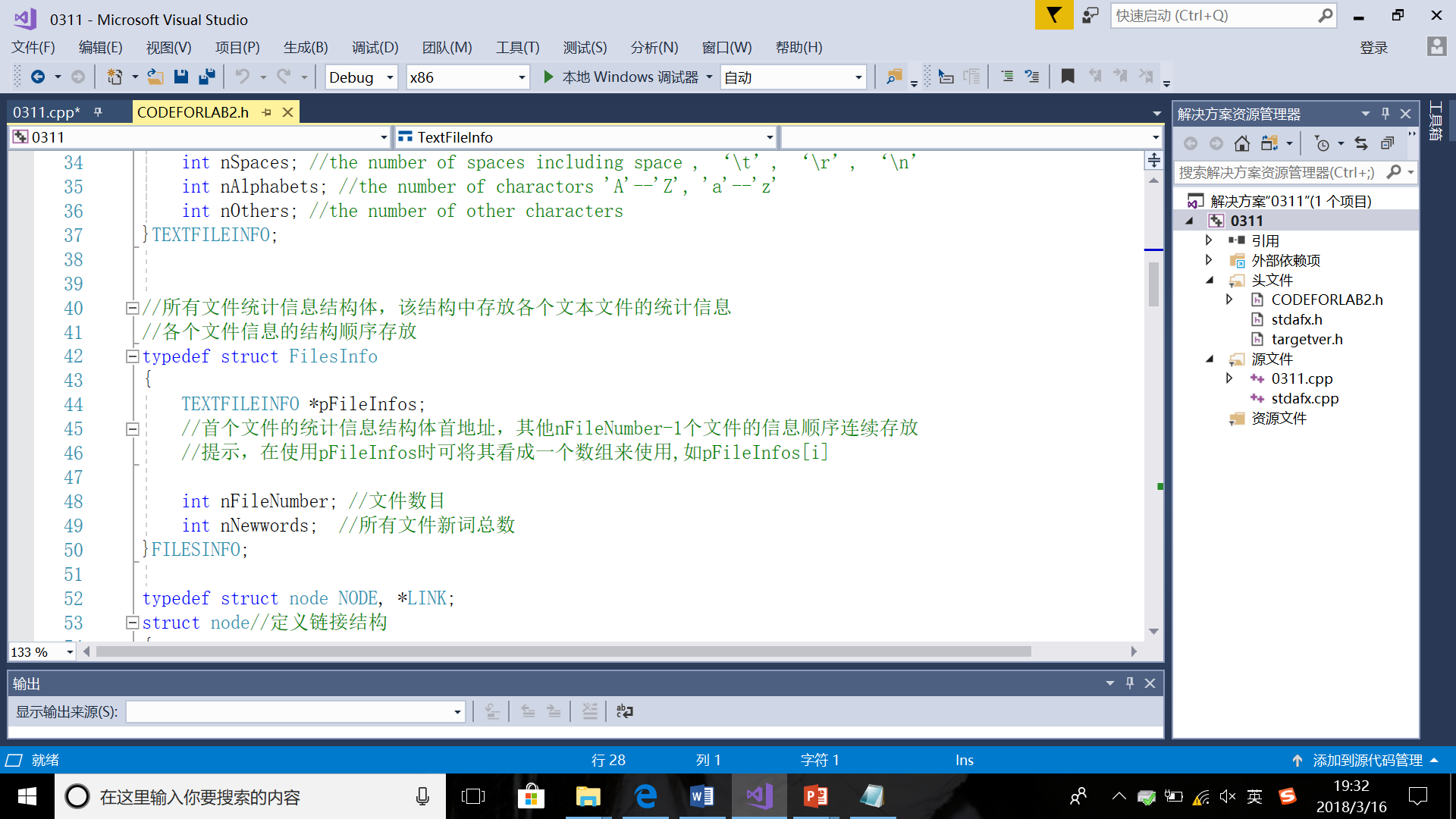
定义了一个struct TextFileInfo类型的结构体，该结构体包含实验中每个文件的所有信息，包含用户所给的文件名，所需要同级的单词数，数字数，空白数和其他字符的个数以及文件的大小，并将这个结构体起名为“TEXTFILEINFO”。“

图 2.2

定义了一个struct FilesInfo类型的结构体变量，包含一个TEXTFILEINFO类型的指针，文件数目和文件新词总数。由后续的AllocateSpaceForTextFilesInfo 可知，pFileInfos指针指向长度为nFileNumber的TEXTFILEINFO类型的结构体数组的首地址。

1. 能否熟练掌握并倾底理解本题中存储与读取任一个文件并统计结果的方法。并请在实验报告中说明typedef的作用。

typedef的作用：

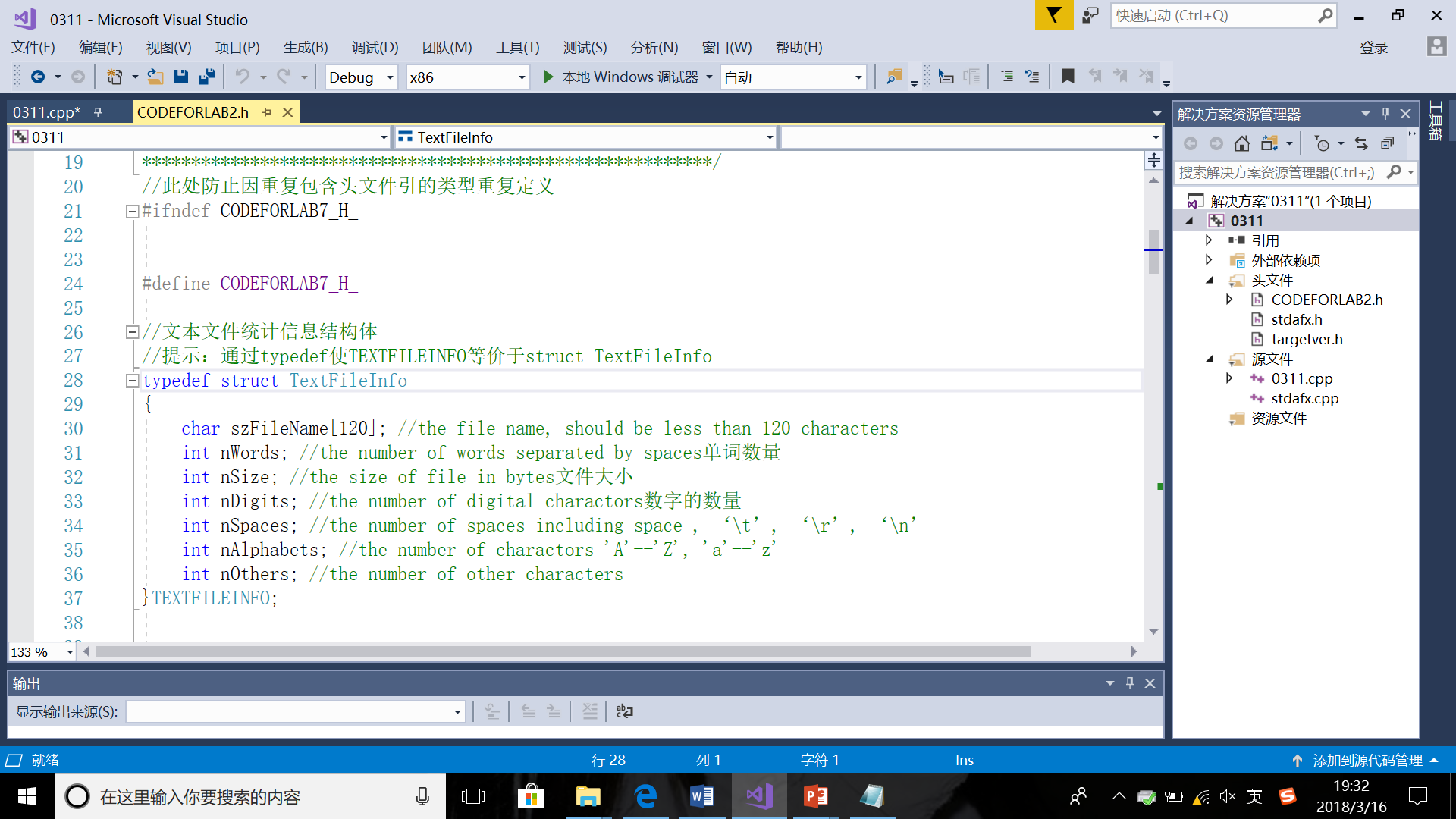


图 2.3

自定义了TEXTFILEINFO类型的结构体变量，该类型的变量用来描述每个文件的信息，包括用户所给的文件名，所需要统计的单词数，数字数，空白数和其他字符的个数以及文件的大小。

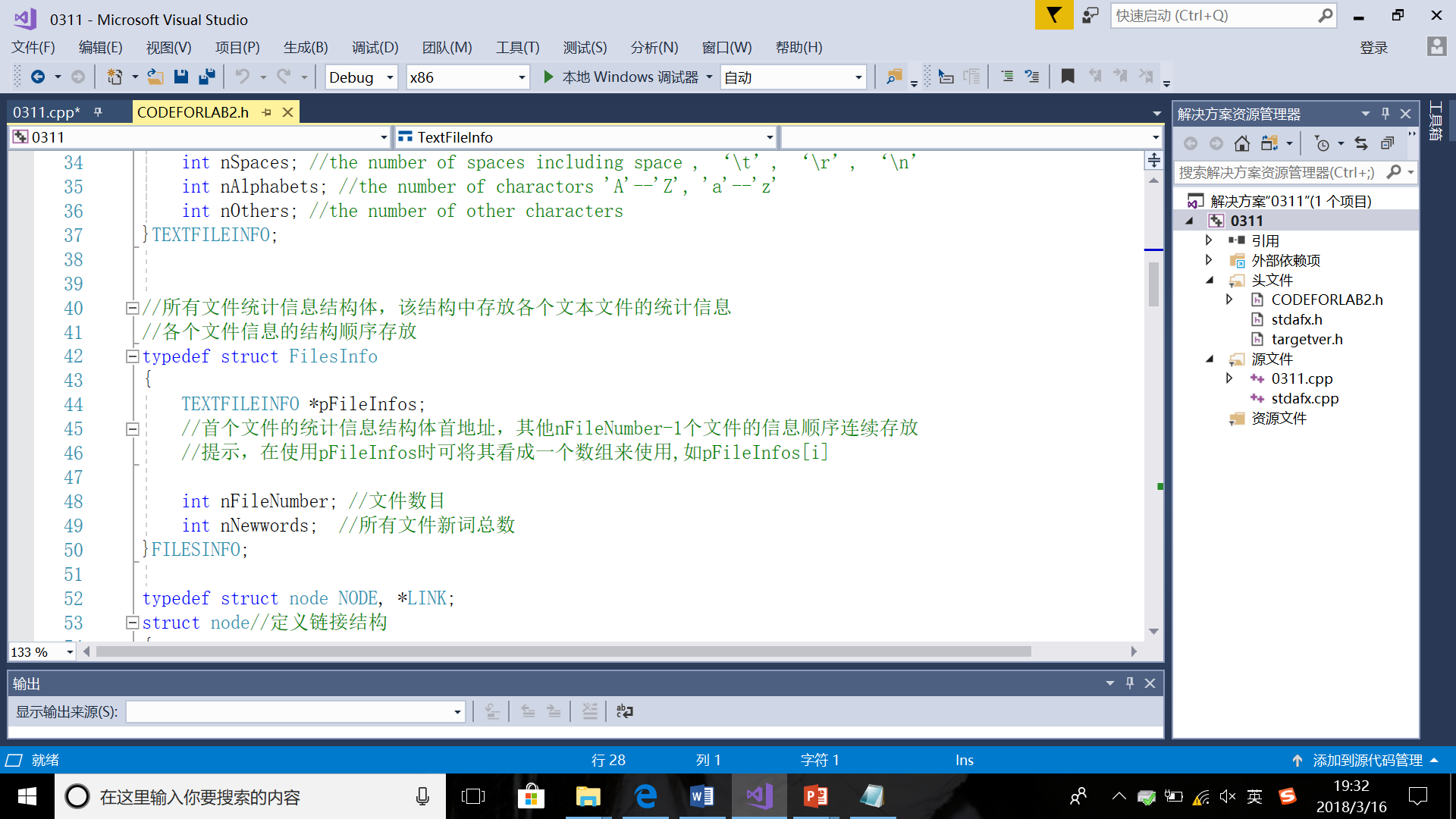


图 2.4

自定义了FILESINFO类型的结构体，该类型变量用来描述整个实验的所有信息。

例如，FILESINFO allfilesinfo; 该语句定义了FILESINFO类型的结构体变量allfilesinfo，其中有三个成员：pFileInfos指向一个TEXTFILEINFO类型的结构体数组，该数组中的每个元素都存储了一个文件的所有信息；nFileNumber是实验中所用的文件数目，nNewwords是所有文件的新词总数。

2）阅读函数 AllocateSpaceForTextFilesInfo，请解释英文单词 allocate 和 memory 的含义。查阅资料并在实验报告中说明函数 malloc的具体功能。

allocate:申请，分配。

Memory:内存

功能：在内存的动态分配区分配一个长度为（size）的连续存储空间，并将此存储空间的地址作为函数值返回，该返回值是一个指向分配起始地址的指针。

具体功能：

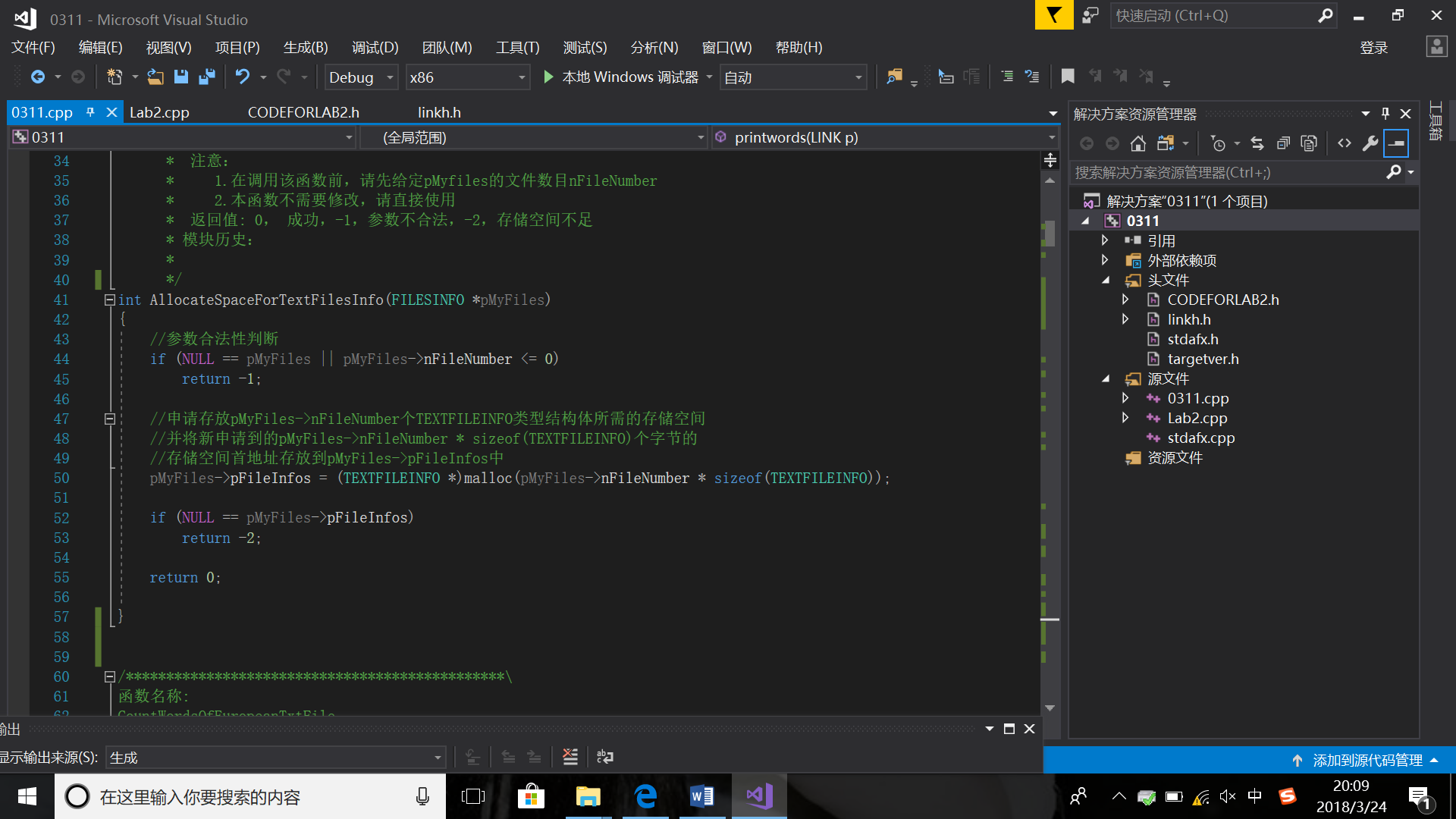


图 2.5

申请存放pMyFiles->nFileNumber个TEXTFILEINFO类型结构体所需的存储空间，并将新申请到的pMyFiles->nFileNumber \* sizeof(TEXTFILEINFO)个字节的存储空间首地址存放到pMyFiles->pFileInfos中。

3）阅读函数 CountWordsOfEuropeanTxtFile，画出该函数流程图，查阅资料并在实验报告中说明其中的函数fopen或fclose的具体功能。

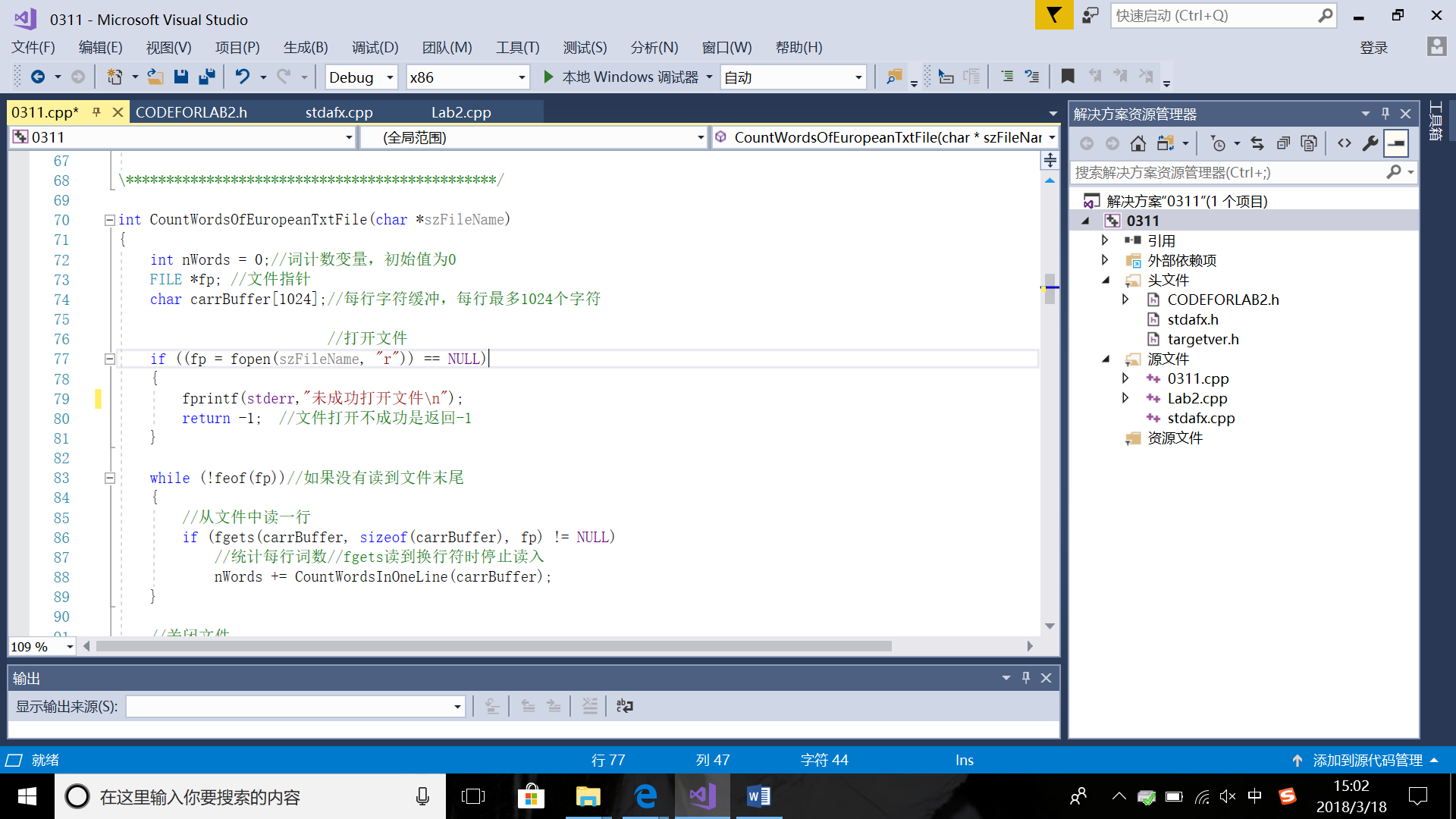


图 2.6

Fopen():如图，将pFilesInfo指针所指向的pFileInfos第i个成员的szFileName的地址传给改指针。用fopen()函数打开该文件。Fopen()函数返回一个文件指针，将其赋值给fp，建立起fp和所需要读取的文件之间的联系。Fopen(szFilename,”r”)说明被打开文件的名字是szFilename所指向的字符串，并以只读的方式打开。

Fclose():用fclose()函数关闭文件，废除fp和该文件的联系。

4）阅读CodeForLab2.cpp的各个函数的功能说明，补全代码实现如下各个函数的功能：

PrepareBasicInfoOfFiles

CountFilesInfo

CountDifferentCharactersInFile

TestCountFilesInfo

英文单词的统计功能，请调用上一小题完成的函数 CountWordsInOneLine

5）调用 TestCountFilesInfo验证各功能正确性。

6）阅读函数 ExportFilesInfoToWebPage，查阅资料并在实验报告中说明函数fprintf 的功能。

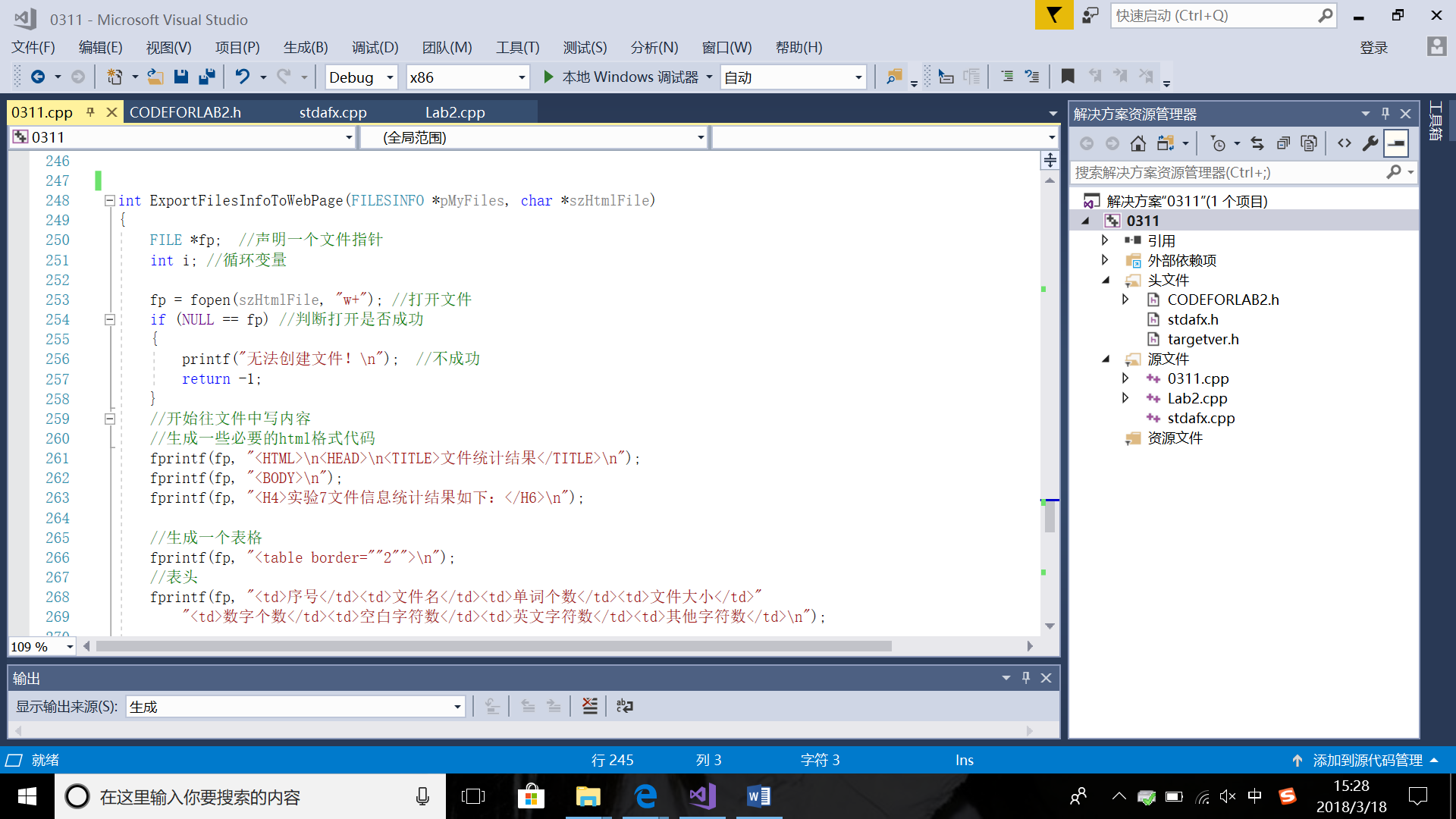


图 2.7

此为fprintf()函数

此为函数的实现结果

由上图可知，fprintf()函数用于将双引号中的内容输出到网页上。

3.思考：

1. Testcountfilesinfo()函数：

（1）

int PrepareBasicInfoOfFiles(FILESINFO \*pFilesInfo)

{

//请在此往后补充代码

printf("请输入文件的个数");

scanf("%d", &pFilesInfo->nFileNumber); //给定实验用文本文件的个数

AllocateSpaceForTextFilesInfo(pFilesInfo); //申请给定个数的文件所需要的空间

printf("请输入文件名");

int i = 0;

for (; i < pFilesInfo->nFileNumber; i++)

{

scanf("%s", **pFilesInfo->pFileInfos[i].szFileName**); //读取文件名

}

return 0;

}

**/\*pFilesInfo->pFileInfos[i].szFileName---**关于读取文件名的方法

pFilesInfo是指针，引用其成员需要用->，其成员pFileInfos也是一个指针，但是指向结构体数组的指针，这个数组的内存是通过动态分配而来的。此时pFileInfos相当于一个数组名，数组名[i]表示第i个数组元素的首地址，.szFileName表示这个结构体中的字符数组名，也就是字符串的首地址。\*/

（2）

int CountWordsOfEuropeanTxtFile(char \*szFileName)

{

int nWords = 0;//词计数变量，初始值为0

FILE \*fp; //文件指针

char carrBuffer[1024];//每行字符缓冲，每行最多1024个字符

//打开文件

if ((fp = fopen(szFileName, "r")) == NULL)

{

fprintf("未成功打开文件\n");

return -1; //文件打开不成功是返回-1

}

while (**!feof(fp)**)//如果没有读到文件末尾

{

//从文件中读一行

if (fgets(carrBuffer, sizeof(carrBuffer), fp) != NULL)

//统计每行词数//fgets读到换行符时停止读入

nWords += CountWordsInOneLine(carrBuffer);

}

//关闭文件

fclose(fp);

return nWords;

}

/\*关于feof(FILE \*fp)函数

feof ( )函数，读取文件的最后一个字符以后，C 语言的feof ( ) 函数依然返回 0，表明没有到达文件结尾；只有当fgetc ( ) 向后再读取一个字符（即越过最后一个字符），feof()才会返回一个非零值，表示到达文件结尾。\*/

（3）总结

对于Testcountfilesinfo()函数，主要考察对于结构体的理解以及对结构体成员的引用、结构体成员作为函数参数的使用。

1. ALinkMaker()函数

（1）

LINK makenode(char w[])

{

**LINK p = (LINK)malloc(sizeof(NODE));** //申请结点内存

**if (p != NULL)**  //申请成功

{

**strncpy(p->word, w, 20);** //将w的长度为MAXLEN的内容复制到p->word中

p->count = 1; //改变结点计数器

p->next = NULL; //makenode的结点总是尾结点（新的结点）

}

return p;

}

/\*

LINK p = (LINK)malloc(sizeof(NODE)); 注意此处的sizeof()的参数一定要写成结构体类型，而不是结构体类型的指针。

if (p != NULL) 注意此处一定要对是否申请内存成功做出判断

关于strncpy（）函数：

1.Strncpy()是strcpy()的扩展，使用时包含在头文件#include<string.h>中

2.函数原型：strncpy(char \*s1,const char \*s2,int n);

其中：

目标字符s1，源字符s2，拷贝长度n

将s2所指的字符串的前n个长度的字符放到s1所指向的字符串中，并将s1原有的前n个字符覆盖。

\*/