

# 题目：词法分析程序的设计与实现

## 实验内容：

设计并实现 C 语言的词法分析程序，要求如下。

- (1)、可以识别出用 C 语言编写的源程序中的每个单词符号，并以记号的形式输出每个单词符号。
- (2)、可以识别并读取源程序中的注释。
- (3)、可以统计源程序汇总的语句行数、单词个数和字符个数，其中标点和空格不计算为单词，并输出统计结果
- (4)、检查源程序中存在的错误，并可以报告错误所在的行列位置。
- (5)、发现源程序中存在的错误后，进行适当的恢复，使词法分析可以继续进行，通过一次词法分析处理，可以检查并报告源程序中存在的所有错误。

## 实验要求：

方法 1：采用 C/C++ 作为实现语言，手工编写词法分析程序。

方法 2：通过编写 LEX 源程序，利用 LEX 软件工具自动生成词法分析程序。

## 算法思路：

首先通过遍历，统计行号和字符数，并记录所有的注释，同时每一行作为一句添加'\0'符号。

其次，再次读取，利用一个字符数组作为 buffer 保存一行的数据，在对其中的数据进行处理，完成之后再读下一行，跳过注释。

对于整行数据的处理，依靠空格将其分成单个单词再具体处理。

对于查错问题，我只能根据一些规则判定有错并记录。

班级：2015211310

学号：2015211406

姓名：任学彬

编译原理——词法分析实验报告

源代码：

```
#include<iostream>
#include<fstream>
#include<vector>
#include<string>
#include<map>
#include<cctype>
using namespace std;
```

```
const int MAX_INPUT_BUFFER = 512;//默认输入缓存区
```

```
ifstream file;//打开文件
```

```
//vector<string>keyword;//预先记录关键字
```

```
map<string, int> keyword = {
```

```
    { "auto",0 },
    { "double",0 },
    { "int",0 },
    { "struct",0 },
    { "break",0 },
    { "else",0 },
    { "long",0 },
    { "switch",0 },
    { "case",0 },
    { "enum",0 },
    { "register",0 },
    { "typedef",0 },
    { "char",0 },
    { "extern",0 },
    { "return",0 },
    { "union",0 },
    { "const",0 },
    { "float",0 },
```

```
    { "short",0 },
    { "unsigned",0 },
    { "continue",0 },
    { "for",0 },
    { "signed",0 },
    { "void",0 },
    { "default",0 },
    { "goto",0 },
    { "sizeof",0 },
    { "volatile",0 },
    { "do",0 },
    { "if",0 },
    { "while",0 },
    { "static",0 }
```

```
};
```

```
vector<string>comment;//存储注释
```

```
map<string, int> id;//标识符
```

```
vector<string>pre;//预处理文件
```

```
bool in_comment = false;//标识 此行是否在注释中
```

```
int sum_char = 0;//字符数
```

```
int sum_word = 0;//单词数
```

```
int sum_line = 0;//行数
```

```
int current_line = 0;//当前行数
```

```
char buffer[MAX_INPUT_BUFFER];//缓存区
```

```
int big_bracket = 0;//大括号
```

```
int small_bracket = 0;//小括号
```

void sum(void)//计算字符数 行数 存注释

```
{
    file.open("data.c");
    int state = 0;//状态转换 0:正常 1:输入一个/ 2:输入/*
    char temp;//暂存
    string temp_comment;//暂存注释
    file.seekg(0);//文件指针回到头
    while (!file.eof())//遍历全文
    {
        switch (state)
        {
            case 0:
                file.get(temp);
                sum_char++;//增加字符
                if ('\n' == temp)
                    sum_line++;//增加行
                if ('/' == temp)
                    state = 1;
                break;
            case 1://前一个字符是 /
                file.get(temp);
                sum_char++;//增加字符
                if ('/' == temp)//单行注释
                {
                    temp_comment.clear();//清空注释缓存
                    while ('\n' != temp)//输完整行
                    {
                        file.get(temp);
                        sum_char++;//增加字符
                        temp_comment.append(1, temp);//添加
```

注释

注释记录表

存入注释表

```

        }
        sum_line++;//行增加
        comment.push_back(temp_comment);//添 加 到
        state = 0;//状态回归
    }
    else
        if ('*' == temp)//多行注释
        {
            temp_comment.clear();
            state = 2;
        }
        else//不是注释
        {
            if ('\n' == temp)
                sum_line++;//增加行
            state = 0; //状态回归
        }
        break;
    case 2:
        file.get(temp);
        sum_char++;//增加字符
        if ('*' == temp)
        {
            file.get(temp);//再取一个
            sum_char++;//增加字符
            if ('/' == temp)//结束多行注释
            {
                comment.push_back(temp_comment);//
                state = 0;
            }
        }
    }
```

班级：2015211310

学号：2015211406

姓名：任学彬

编译原理——词法分析实验报告

else//还在注释中

```
{
    temp_comment.append(1, '*');
    temp_comment.append(1, temp);
    if ('\n' == temp)
        sum_line++;
}
}
else
{
    if (file.eof())//当注释到尾时
    {
        if (!temp_comment.empty())//不为空时
        {
            comment.push_back(temp_comment);

            return;
        }
    }
    if ('\n' == temp)
        sum_line++;
    temp_comment.append(1, temp);
}
}
break;

default:
    break;
}
}
```

存入最后一个注释

void word\_analyse(void)

```
{
    string temp_word;//暂存单词
    char *ptr = buffer;//处理指针

    while (in_comment)//多行注释问题
    {
        if ('\0' == *ptr)
            return;
        if ((*' == *ptr) && ('/' == *(ptr + 1)))
        {
            ptr += 2;
            in_comment = false;
        }
        ptr++;
    }
    while ('\0' != *ptr)
    {
        if (' ' == *ptr)//消空格
            ptr++;
        else if (isalpha(*ptr) || '_' == *ptr)//关键字和标识符
        {
            temp_word.clear();
            while (isalnum(*ptr) || '_' == *ptr)
            {
                temp_word.append(1, *ptr);
                ptr++;
            }
            //空格或其他字符时弹出
            sum_word++;
        }
    }
}
```

班级：2015211310

学号：2015211406

姓名：任学彬

编译原理——词法分析实验报告

```
map<string, int>::iterator ltr;
ltr = keyword.find(temp_word);
if (ltr != keyword.end()) {
    ltr->second++;
    cout << endl << "第" << sum_word << "个单词： "
    << " 保留字： " << temp_word << endl;
}
else {
    ltr = id.find(temp_word);
    if (ltr != id.end())
        ltr->second++;
    else
        id.insert(pair<string, int>(temp_word, 1));
    cout << endl << "第" << sum_word << "个单词： "
    << " 标识符： " << temp_word << endl;
}

else if (isdigit(*ptr))//数字
{
    temp_word.clear();
    while (isdigit(*ptr))
    {
        temp_word.append(1, *ptr);
        ptr++;
    }
    if (isalpha(*ptr) || '_' == *ptr)
    {
        while (' ' == *ptr && '\0' == *ptr)
        {
            temp_word.append(1, *ptr);
            ptr++;
        }
    }
}
```

```

    }
    sum_word++;

    cout << endl << "第" << sum_word << "个单词： "
    << " 非法字符： " << temp_word << endl;
    cout << "error:在第" << current_line << "行,单词"
    << temp_word << "命名非法！" << endl;
}
else
{
    if ('.' == *ptr || 'E' == *ptr || 'e' == *ptr)//小数和指
    {
        temp_word.append(1, *ptr);
        ptr++;
        while (isdigit(*ptr))
        {
            temp_word.append(1, *ptr);
            ptr++;
        }
    }
    sum_word++;

    cout << endl << "第" << sum_word << "个单词： "
    << " 无符号数： " << temp_word << endl;
}
else if ('#' == *ptr)//预处理文件
{
    while ('\0' != *ptr)
    {
        temp_word.append(1, *ptr);
        ptr++;
    }
}
```

数形式

班级：2015211310

学号：2015211406

姓名：任学彬

编译原理——词法分析实验报告

```
    }
    pre.push_back(temp_word);
}

else if ("" == *ptr)//字符串
{
    temp_word.clear();
    ptr++;
    while ("" != *ptr)
    {
        temp_word.append(1, *ptr);
        ptr++;
    }
    ptr++;
    sum_word++;
    cout << endl << "第" << sum_word << "个单词: " << " 字
字符串: " << temp_word << endl;
}

else if ('+' == *ptr)
{
    temp_word.clear();
    if ('+' == *(ptr + 1))
    {
        temp_word = "++";

        map<string, int>::iterator ltr;
        ltr = id.find(temp_word);
        if (ltr != id.end())
            ltr->second++;
        else
            id.insert(pair<string, int>(temp_word, 1));
    }
}
```

```
    sum_word++;
    cout << endl << "第" << sum_word << "个单词: "
<< " 加等于号: " << temp_word << endl;
    ptr += 2;
}
else if ('=' == *(ptr + 1))
{
    temp_word = "+=";

    map<string, int>::iterator ltr;
    ltr = id.find(temp_word);
    if (ltr != id.end())
        ltr->second++;
    else
        id.insert(pair<string, int>(temp_word, 1));

    sum_word++;
    cout << endl << "第" << sum_word << "个单词: "
<< " 自加号: " << temp_word << endl;
    ptr += 2;
}
else if (isdigit(*(ptr + 1)))//有符号数
{
    temp_word = "+";
    for (int j = 1; isdigit(*(ptr + j)); j++)
        temp_word.append(1, *(ptr + j));

    sum_word++;
    cout << endl << "第" << sum_word << "个单词: "
<< " 有符号数: " << temp_word << endl;
    ptr += 2;//推进
}
```

班级：2015211310

学号：2015211406

姓名：任学彬

编译原理——词法分析实验报告

```
else
{
    temp_word = "+";

    map<string, int>::iterator ltr;
    ltr = id.find(temp_word);
    if (ltr != id.end())
        ltr->second++;
    else
        id.insert(pair<string, int>(temp_word, 1));

    sum_word++;
    cout << endl << "第" << sum_word << "个单词： "
    << " 加号： " << temp_word << endl;
    ptr += 2; //推进
}

else if ('-' == *ptr)
{
    temp_word.clear();
    if ('-' == *(ptr + 1))
    {
        temp_word = "--";

        map<string, int>::iterator ltr;
        ltr = id.find(temp_word);
        if (ltr != id.end())
            ltr->second++;
        else
            id.insert(pair<string, int>(temp_word, 1));
```

```
sum_word++;
    cout << endl << "第" << sum_word << "个单词： "
    << " 自减号： " << temp_word << endl;
    ptr += 2;
}
else if ('=' == *(ptr + 1))
{
    temp_word = "-=";

    map<string, int>::iterator ltr;
    ltr = id.find(temp_word);
    if (ltr != id.end())
        ltr->second++;
    else
        id.insert(pair<string, int>(temp_word, 1));

    sum_word++;
    cout << endl << "第" << sum_word << "个单词： "
    << " 减等于号： " << temp_word << endl;
    ptr += 2;
}
else if ('>' == *(ptr + 1))
{
    temp_word = "->";

    map<string, int>::iterator ltr;
    ltr = id.find(temp_word);
    if (ltr != id.end())
        ltr->second++;
    else
        id.insert(pair<string, int>(temp_word, 1));
```

班级：2015211310

学号：2015211406

姓名：任学彬

编译原理——词法分析实验报告

```
sum_word++;
cout << endl << "第" << sum_word << "个单词： "
<< " 指针号： " << temp_word << endl;
ptr += 2;
}
else if (isdigit(*(ptr + 1)))//有符号数
{
    temp_word = "-";
    for (int j = 1; isdigit(*(ptr + j)); j++)
        temp_word.append(1, *(ptr + j));

    sum_word++;
    cout << endl << "第" << sum_word << "个单词： "
    << " 有符号数： " << temp_word << endl;
    ptr += 2;//推进
}
else
{
    temp_word = "-";

    map<string, int>::iterator ltr;
    ltr = id.find(temp_word);
    if (ltr != id.end())
        ltr->second++;
    else
        id.insert(pair<string, int>(temp_word, 1));

    sum_word++;
    cout << endl << "第" << sum_word << "个单词： "
    << " 减号： " << temp_word << endl;
    ptr += 2;//推进
}
}
```

```
else if ('*' == *ptr)
{
    temp_word.clear();
    if ('=' == *(ptr + 1))
    {
        temp_word = "*=";

        map<string, int>::iterator ltr;
        ltr = id.find(temp_word);
        if (ltr != id.end())
            ltr->second++;
        else
            id.insert(pair<string, int>(temp_word, 1));

        sum_word++;
        cout << endl << "第" << sum_word << "个单词： "
        << " 乘等于号： " << temp_word << endl;
        ptr += 2;
    }
    else
    {
        temp_word = "*";

        map<string, int>::iterator ltr;
        ltr = id.find(temp_word);
        if (ltr != id.end())
            ltr->second++;
        else
            id.insert(pair<string, int>(temp_word, 1));

        sum_word++;
    }
}
```



班级：2015211310

学号：2015211406

姓名：任学彬

编译原理——词法分析实验报告

```
cout << endl << "第" << sum_word << "个单词： "
<< " 乘号： " << temp_word << endl;

    ptr += 2; //推进
}
}
else if ('/' == *ptr)
{
    temp_word.clear();
    if ('=' == *(ptr + 1))
    {
        temp_word = "/=";

        map<string, int>::iterator ltr;
        ltr = id.find(temp_word);
        if (ltr != id.end())
            ltr->second++;
        else
            id.insert(pair<string, int>(temp_word, 1));

        sum_word++;
        cout << endl << "第" << sum_word << "个单词： "
        << " 除等于号： " << temp_word << endl;
        ptr += 2;
    }
    else if ('/' == *(ptr + 1))
        return; //行注释直接跳出
    else if ('*' == *(ptr + 1)) //多行注释
    {
        in_comment = true;
        ptr += 2;
        while ('\0' != *ptr) //此行结束注释是否也结束
        {
```

```
if (('*' == *ptr) && ('/' == *(ptr + 1)))
{
    in_comment = false;
    break;
}
ptr++;
}
}
else
{
    temp_word = "/";

    map<string, int>::iterator ltr;
    ltr = id.find(temp_word);
    if (ltr != id.end())
        ltr->second++;
    else
        id.insert(pair<string, int>(temp_word, 1));

    sum_word++;
    cout << endl << "第" << sum_word << "个单词： "
    << " 除号： " << temp_word << endl;
    ptr += 2; //推进
}
}

else if ('=' == *ptr)
{
    temp_word.clear();
    if ('=' == *(ptr + 1))
    {
        temp_word = "==";
```

班级：2015211310

学号：2015211406

姓名：任学彬

编译原理——词法分析实验报告

```
{
    map<string, int>::iterator ltr;
    ltr = id.find(temp_word);
    if (ltr != id.end())
        ltr->second++;
    else
        id.insert(pair<string, int>(temp_word, 1));

    sum_word++;
    cout << endl << "第" << sum_word << "个单词： "
    << " 相等比较号： " << temp_word << endl;
    ptr += 2;
}
else
{
    temp_word = "=";

    map<string, int>::iterator ltr;
    ltr = id.find(temp_word);
    if (ltr != id.end())
        ltr->second++;
    else
        id.insert(pair<string, int>(temp_word, 1));

    sum_word++;
    cout << endl << "第" << sum_word << "个单词： "
    << " 赋值号： " << temp_word << endl;
    ptr += 2; //推进
}
}

else if ('>' == *ptr)
```

```
{
    temp_word.clear();
    if ('=' == *(ptr + 1))
    {
        temp_word = ">=";

        map<string, int>::iterator ltr;
        ltr = id.find(temp_word);
        if (ltr != id.end())
            ltr->second++;
        else
            id.insert(pair<string, int>(temp_word, 1));

        sum_word++;
        cout << endl << "第" << sum_word << "个单词： "
        << " 大于等于号： " << temp_word << endl;
        ptr += 2;
    }
    else
    {
        temp_word = ">";

        map<string, int>::iterator ltr;
        ltr = id.find(temp_word);
        if (ltr != id.end())
            ltr->second++;
        else
            id.insert(pair<string, int>(temp_word, 1));

        sum_word++;
        cout << endl << "第" << sum_word << "个单词： "
        << " 大于号： " << temp_word << endl;
    }
}
```

班级：2015211310

学号：2015211406

姓名：任学彬

编译原理——词法分析实验报告

ptr += 2;//推进

id.insert(pair<string, int>(temp\_word, 1));

}

}

else if ('<' == \*ptr)

{

temp\_word.clear();

if ('=' == \*(ptr + 1))

{

temp\_word = "<=";

map<string, int>::iterator ltr;

ltr = id.find(temp\_word);

if (ltr != id.end())

ltr->second++;

else

id.insert(pair<string, int>(temp\_word, 1));

sum\_word++;

cout << endl << "第" << sum\_word << "个单词： "

<< " 小于等于号： " << temp\_word << endl;

ptr += 2;

}

else

{

temp\_word = "<";

map<string, int>::iterator ltr;

ltr = id.find(temp\_word);

if (ltr != id.end())

ltr->second++;

else

sum\_word++;

cout << endl << "第" << sum\_word << "个单词： "  
<< " 小于号： " << temp\_word << endl;

ptr += 2;//推进

}

}

else if ('!' == \*ptr)

{

temp\_word.clear();

if ('=' == \*(ptr + 1))

{

temp\_word = "!=";

map<string, int>::iterator ltr;

ltr = id.find(temp\_word);

if (ltr != id.end())

ltr->second++;

else

id.insert(pair<string, int>(temp\_word, 1));

sum\_word++;

cout << endl << "第" << sum\_word << "个单词： "  
<< " 不等于号： " << temp\_word << endl;

ptr += 2;

}

else

{

temp\_word = "!=";

班级：2015211310

学号：2015211406

姓名：任学彬

编译原理——词法分析实验报告

```
map<string, int>::iterator ltr;
ltr = id.find(temp_word);
if (ltr != id.end())
    ltr->second++;
else
    id.insert(pair<string, int>(temp_word, 1));

sum_word++;
cout << endl << "第" << sum_word << "个单词： "
<< " 取反号： " << temp_word << endl;
ptr += 2;//推进
}
}

else if (':' == *ptr || '(' == *ptr || ')' == *ptr || ';' == *ptr || '{' == *ptr
|| '}' == *ptr || ',' == *ptr
|| '[' == *ptr || ']' == *ptr || '\0' == *ptr || '\n' == *ptr)//标点不算单词
{
    if '(' == *ptr)
        small_bracket++;//查错
    if ')' == *ptr)
        small_bracket--;//查错
    if '{' == *ptr)
        big_bracket++;//查错
    if '}' == *ptr)
        big_bracket--;//查错
    ptr++;//推进
}
else
{
    cout << endl << "错误:在第" << current_line << "行有错误：
```

```
" << *ptr << endl;
    ptr++;
}
}

void analyse(void)
{
    char *p = buffer;
    file.seekg(0);
    while (!file.eof())
    {
        file.get(*p);
        if ('\n' == *p)//一句
        {
            *(p + 1) = '\0';
            *(p + 2) = '\0';
            current_line++;
            word_analyse();
            if (small_bracket != 0)
            {
                cout << endl << "错误:在第" << current_line << "行
小括号不匹配" << endl;
                small_bracket = 0;
            }
            p = buffer;//重新填充
        }
        else
            p++;
    }
    if (big_bracket != 0)
        cout << endl << "错误:大括号不匹配！" << endl;
```

班级：2015211310

}

学号：2015211406

void show(void)

{

cout << endl;

map<string, int>::iterator ltr;

for (ltr = keyword.begin(); ltr != keyword.end(); ltr++)

if (ltr->second != 0)

cout << ltr->first << "\t" << ltr->second << endl;

for (ltr = id.begin(); ltr != id.end(); ltr++)

cout << ltr->first << "\t" << ltr->second << endl;

cout << endl << "单词总数:" << sum\_word << endl;

cout << endl << "字符总数:" << sum\_char << endl;

cout << endl << "总行数 is:" << sum\_line << endl;

cout << "头文件有:" << endl;

for (int i = 0; i < pre.size(); i++)

{

cout << pre.at(i);

}

cout << "注释有:" << endl;

for (int i = 0; i < comment.size(); i++)

{

cout << comment.at(i);

姓名：任学彬

}

cout << endl;

}

int main(void)

{

file.open("data.c");

if (!file)

cout << endl << "找不到文件！" << endl;

else

{

//开始处理

analyse();

file.close();

sum();//统计行数 字节数 注释

show();//显示统计结果

}

system("pause");

return 0;

}

结果分析：

正确程序测试

```
1 #include<stdio.h>
2 /*
3 测试数据
4 */
5 int main(void)
6 {
7     printf("hello world!");
8
9     return 0;
10 }
11
```

第1个单词: 保留字: int  
第2个单词: 标识符: main  
第3个单词: 保留字: void  
第4个单词: 标识符: printf  
第5个单词: 字符串: hello world!  
第6个单词: 保留字: return  
第7个单词: 无符号数: 0

各符号及出现次数统计如下:

int	1
return	1
void	1
main	1
printf	1

单词总数:7  
字符总数:93  
总行数is:11  
头文件有:  
#include<stdio.h>  
注释有:  
测试数据  
请按任意键继续. . .

字符数包括空格

复杂程序测试

data.c, 由于太多无法截图, 可以自行尝试。

```
1 #include<stdio.h>
2 /*
3 测试注释
4 */
5
6 int main(
7 {
8     char k3; int i = 1;
9     while(i<10)
10     {
11         printf("hello world!"); //打印hello world
12
13         return 0;
14     }
15 }
```

第1个单词: 保留字: int  
第2个单词: 标识符: main  
错误: 在第6行小括号不匹配  
第3个单词: 保留字: char  
第4个单词: 非法字符: 8  
error: 在第8行, 单词8命名非法!  
第5个单词: 标识符: k3  
第6个单词: 保留字: int  
第7个单词: 标识符: i  
第8个单词: 赋值号: =  
第9个单词: 无符号数: 1  
错误: 在第9行有错误: @  
第10个单词: 保留字: while  
第11个单词: 标识符: i  
第12个单词: 小于号: <

第12个单词: 小于号: <  
第13个单词: 无符号数: 0  
第14个单词: 标识符: printf  
第15个单词: 字符串: hello world!  
第16个单词: 保留字: return  
第17个单词: 无符号数: 0  
错误: 大括号不匹配!

各符号及出现次数统计如下:

char	1
int	2
return	1
while	1
<	1
=	1
i	2
k3	1
main	1
printf	1

单词总数:17  
字符总数:149

C:\Users\42444\Documents\Visual S C:\Users\42444\Documents\Visual Studio 20

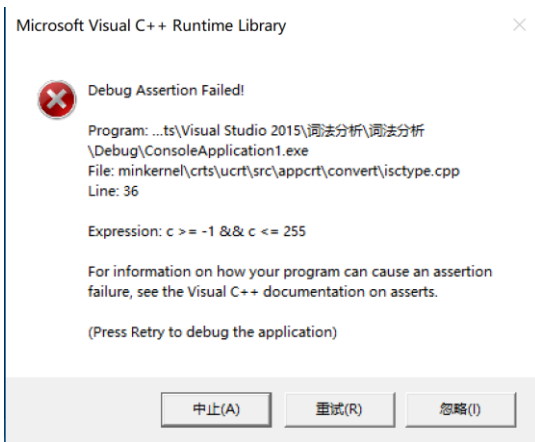
C:\Users\42444\Documents\V

总行数is:16  
头文件有:  
#include<stdio.h>  
注释有:  
测试注释  
打印hello world  
请按任意键继续. . .

字符数包括空格，错误处理方面如命令行第二个红圈所示，将可以构成标识符的作为一个变量

## 实验反思：

- 1、 本次实验可以说我很清楚的了解词法分析的流程，对于这种我们无法直接看到的过程有了概念。
- 2、 代码中的问题有不少，我觉得最大的一点就是这次强迫自己用了 c++ 的独特知识，开始本来想用 `vector` 就够了，但要求中有说需要输出各个字符的出现次数，无奈搜索很久没找到 `vector` 计数（可能是傻了），但是搜到了另一种 c++ 容器——`map`，而这种 `key-value` 的组合深得我心，而且操作简单，同时据了解 `map` 的搜索效率也很高，所以可以说这次的一大收获就是学会了 `map` 的相关操作。
- 3、 代码中还有不足就是格式上没有采用自动机，尽管我自己花了不少时间画了一个自动机的状态转移图，但真正写的时候。。脑子就抽风了，写着写着就写成了 `if-else` 的分支语句。
- 4、 程序中其实讲实话情况没有考虑的很完美，至少想到去看 C99 的标准但是最后因为时间快到所以就没改，会在课下改完的（嗯）。
- 5、 代码的注释可以看到前面挺详尽但后面。。。。实在不想写了，打代码一有状态打好多，然后就不想打注释了。
- 6、 代码的思路在第一页提到了，感觉和一些同学的不同，因为当时由于选用 `vector` 和 `map` 的问题请教同学的时候居然被说代码逻辑看不懂。。。我觉得我这个很傻瓜的逻辑啊，所以可能以后要想的更详尽更仔细一点。



- 7、 这个问题开始没有出现，但写这个报告的时候就出现了，查阅博客这个问题是 `isalpha` 参数中是无符号数应该强制转换为 `unsigned`，但是之后没有用……后来想起了两个小时之前自己作死的把代码传到 `github` 的时候吧编码格式改了（因为 `github` 上乱码），当改回来的时候就没有问题了……很迷。