

实验二 标识符的识别

17计二许红凯 320170941570

任务介绍:

根据给定源语言的构词规律，从任意字符串中识别出所有的合法标识符

输入:

字符串

输出:

单词符号流，一行一个单词

题目:

设计一个程序，从任意字符串中识别出所有可视为C语言“名字”的子串。注意：

1. 构词结构：以字母打头，后跟任意多个字母、数字的单词；长度 不超过15；不区分大小写；把下划线视为第27个字母
2. 关键字保留，即：语言定义中保留了某些单词用作关键字，程序员不可以将这些单词用作“名字”（变量名、常量名、函数名、标记号等等）

代码思路:

1. 判断字符串s第一个字符是否为字母，若是则存入数组ss中，若不是则下一个字符作为第一个字符循环操作1
2. 判断字符串下一个字符，是否为a-z或A-Z或0-9或_，若是则继续存入数组中，若不是则将数组内容与32个关键字进行比较，若不是关键字则输出数组，否则返回到操作1
3. 以上输出控制输出长度，若长度超15则返回到操作1

程序功能说明

1. 从任意字符串中识别出所有可视为C语言“名字”的子串
2. 判断该“名字”是否合法，即要排除关键字

重点说明

- 1.
2. 用数组来暂存当前识别中正常的连续字符

```
char *ss =(char *) malloc(16);

int ss_len = 0; //暂存数组长度
```

3. 构词结构为长度不超过15，故要控制暂存数组长度为15，若超过15需另输出新字符串

```
//控制输出字符数组的长度
    if(ss_len == 15){
        j = j - 1;
        break; // 跳出循环，直接到输出
    }
```

4. 对于每个合法的暂存数组，其要输出作为字符串时，需在末尾加上'\0'表示字符串结束

```
if(tag == 0){
    ss[ss_len] = '\0';
    printf("%s",ss);
    printf("\n");
}
```

```
//判断是否为关键字
    for(k = 0; k < 32; k++){
        if( strcmp(ss,keyword[k]) == 0){
            tag = 1;
            break;
        }
    }
```

5. 判断某合法字符串是否为“关键字”时，用tag = 1来表示是，故不输出作为“名字”

```
//判断是否为关键字
    for(k = 0; k < 32; k++){
        if( strcmp(ss,keyword[k]) == 0){
            tag = 1;
            break;
        }
    }
```

6. 每处理一个字符串后，重新处理下一阶段前，需清空暂存数组

```
memset(ss, 0, 16); //清空暂存数组
```

```
ss_len = 0;
```