# 转载：http://blog.csdn.net/u013213837/article/details/45644297

scanf是C语言中常用的标准输入输出函数，它的常规用法是scanf（“输入格式”，输入地址），scanf从输入缓冲区取入数据按照指定的“输入格式”存储到指定的“输入地址”。

但是有一类问题通常令人头疼，那就是字符串的读入，此处仅仅讨论使用scanf读取字符串的一些问题。

理论上你只需要定义一个字符指针，然后scanf每次读入字符存储到该指针所指的存储空间，之后指针值自动加1.

所以，我们就有了下面两种读入字符串的办法:

* 1
* 2
* 3
* 4
* 5

1.定义字符数组

#include <stdio.h>

#include <stdlib.h>

int main(void) {

// your code goes here

char str[20];

scanf("%s",str);

printf("%s",str);

return 0;

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11

完美运行，但是得说明一点，这里的字符数组必须事先确定好长度，即得事先吧内存开辟好，不然运行肯定会出错。   
2.定义字符指针

#include <stdio.h>

#include <stdlib.h>

int main(void) {

// your code goes here

char\* str;

scanf("%s",str);

printf("%s",str);

return 0;

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11

scanf既然可以在每次读入一个字符之后，自动为指针+1，那么我们自然而然就想到其实我们是可以定义一个字符指针，然后scanf就往这个字符指针中读取字符，但是其实不然，运行一下就会发现这种情况是错误的，为什么呢？   
因为我们没有为scanf输入的指针申请存储空间，scanf输入的数据必须也只能存放到实现申请好的存储空间，不然输入的字符可能会覆盖掉代码段或者其他一些地方的数据，引发未知的错误。   
现在我们就试一试事先申请内存：

#include <stdio.h>

#include <stdlib.h>

int main(void) {

// your code goes here

char\* str=(char\*)malloc(15\*sizeof(char));

scanf("%s",str);

printf("%s",str);

return 0;

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10

完美运行。   
简单总结一下，那就是如果要使用scanf读入字符串，那就一定要事先为它申请足够的内存空间。