

# 第8章 指针

## ——字符数组与字符指针：

---

字符串的输入和输出

# 逐个字符输入输出字符串

```
#define STR_LEN 80  
char str[STR_LEN+1];
```

```
for (i=0; str[i]!='\0'; i++)  
{  
    putchar(str[i]);  
}  
putchar('\n');
```

一般不用字符串长度控制，如  $i < \text{STR\_LEN}$

# 整体输入输出字符串

```
#define STR_LEN 80  
char str[STR_LEN+1];
```

```
scanf("%s", str);  
printf("%s\n", str);
```

不能输入带空格的字符串

可以输入带空格的字符串

```
gets(str);  
puts(str);
```

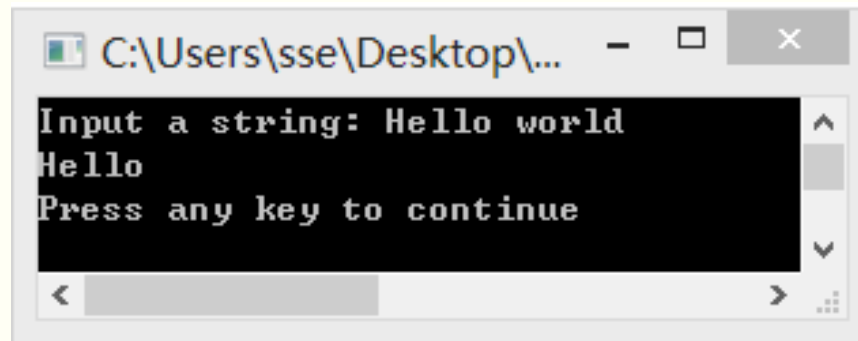
为什么?

# 用scanf输入/输出字符串

```
#include <stdio.h>
#define STR_LEN 80
int main()
{
    char str[STR_LEN+1];

    printf("Input a string:");
    scanf("%s", str);
    printf("%s\n", str);

    return 0;
}
```



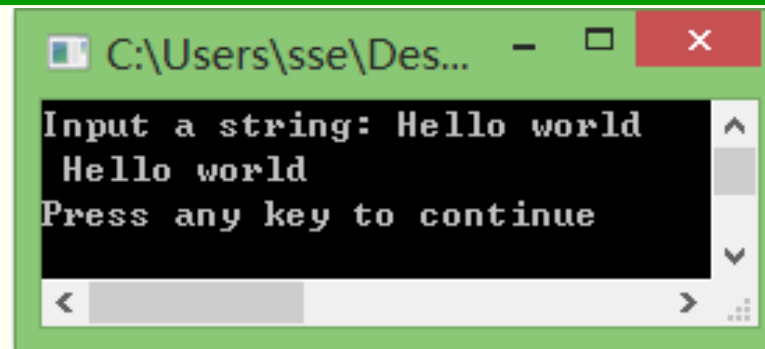
用%d输入数字或%s输入字符串时，在开始读之前会跳过空格、回车或制表符等空白字符，再次遇到这些字符时，系统认为读入结束，因此不能输入带空格的字符串

# 用gets输入/输出字符串

```
#include <stdio.h>
#define STR_LEN 80
int main()
{
    char str[STR_LEN+1];

    printf("Input a string:");
    gets(str);
    printf("%s\n", str);

    return 0;
}
```

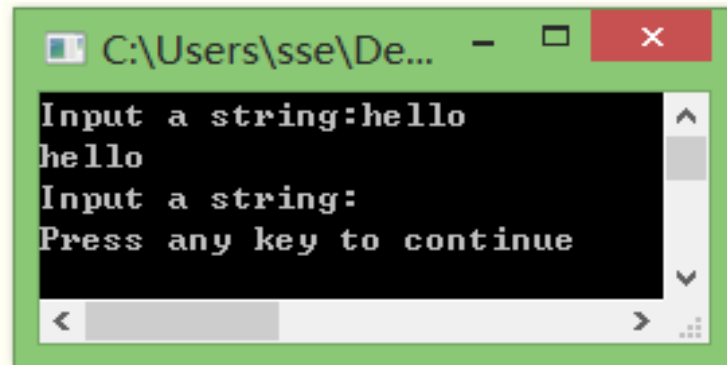


以回车换行作为终止符，可输入带空格的字符串，因为空格和制表符都是字符串的一部分

此外，两个字符串输入函数对回车换行符的处理也不同

# 用gets输入/输出字符串

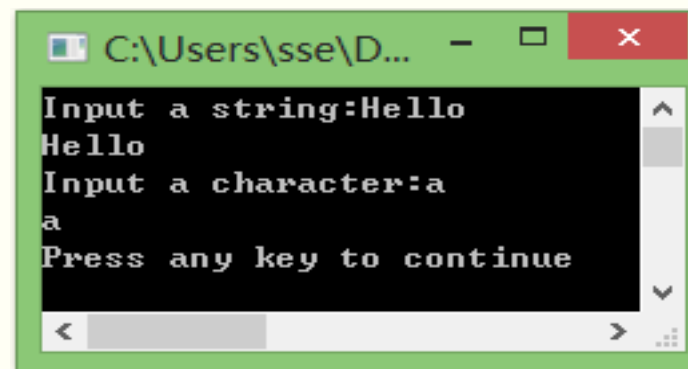
```
#include <stdio.h>
#define STR_LEN 80
int main()
{
    char str[STR_LEN+1];
    printf("Input a string:");
    scanf("%s", str);
    printf("%s\n", str);
    printf("Input a string:");
    gets(str);
    printf("%s\n", str);
    return 0;
}
```



**gets()**将回车从缓冲区读走，但不作为字符串的一部分，用空字符代替

# 用gets输入/输出字符串

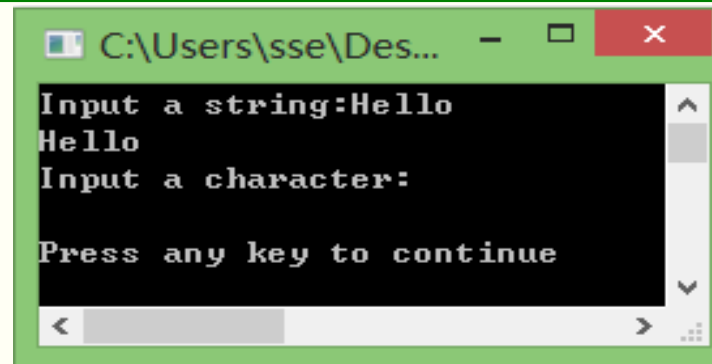
```
#include <stdio.h>
#define STR_LEN 80
int main()
{
    char str[STR_LEN+1], ch;
    printf("Input a string:");
    gets(str);
    printf("%s\n", str);
    printf("Input a character:");
    ch = getchar();
    printf("%c\n", ch);
    return 0;
}
```



gets()将回车从缓冲区读走，  
所以getchar()等待用户输入

# 用scanf输入/输出字符串

```
#include <stdio.h>
#define STR_LEN 80
int main()
{
    char str[STR_LEN+1], ch;
    printf("Input a string:");
    scanf("%s", str);
    printf("%s\n", str);
    printf("Input a character:");
    ch = getchar();
    printf("%c\n", ch);
    return 0;
}
```



**scanf()**不读走回车，回车仍留在缓冲区中，回车会被**getchar()**读走



# 用scanf输入/输出字符串

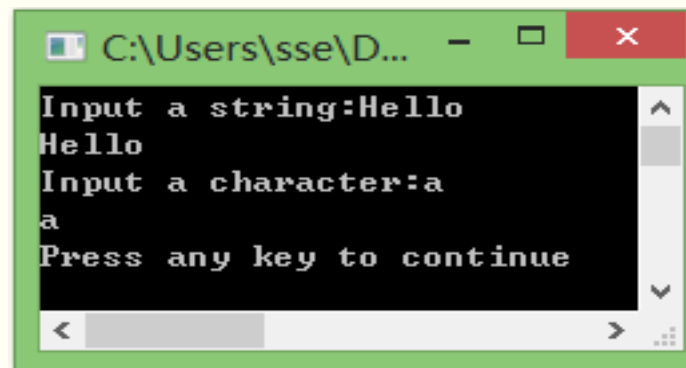
```
#include <stdio.h>
#define STR_LEN 80
int main()
{
    char str[STR_LEN+1], ch;
    printf("Input a string:");
    scanf("%s", str);
    printf("%s\n", str);
    printf("Input a character:");
    getchar();
    ch = getchar();
    printf("%c\n", ch);
    return 0;
}
```

```
scanf(" ");
```

```
scanf(" ");
scanf("%c", &ch);
```

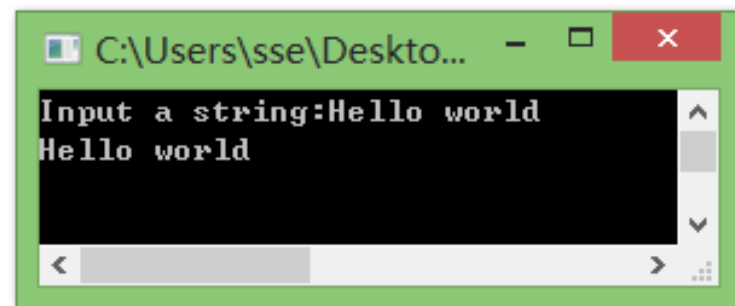
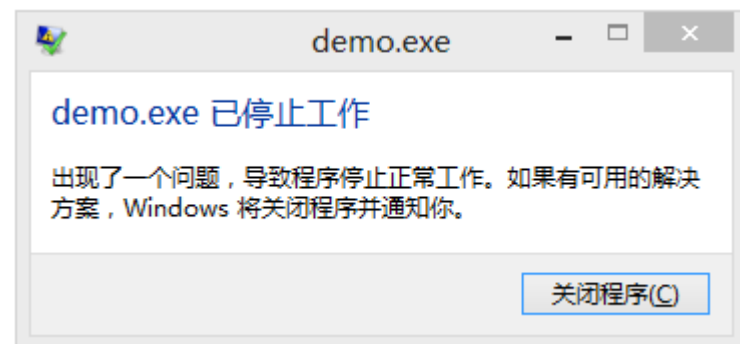
```
scanf(" %c", &ch);
```

清空输入缓冲区中的空白字符



# 用gets输入/输出字符串

```
#include <stdio.h>
#define STR_LEN 80
int main()
{
    char str[STR_LEN+1];
    char *ptr = str;
    printf("Input a string:");
    gets(ptr);
    printf("%s\n", ptr);
    return 0;
}
```



# 讨论

## ■ 见证奇迹的时刻——Mission Impossible

源程序（不要添加任何空格和换行）

```
main(){char*a="main(){char*a=%c%s%c;printf(a,34,a,34);}";printf(a,34,a,34);}
```

运行结果与源代码完全一样，你能解释这个运行结果吗？

```
main(){char*a="main(){char*a=%c%s%c;printf(a,34,a,34);}";  
printf(a,34,a,34);}
```

