<https://blog.csdn.net/qiqiaiairen/article/details/52317570>

头文件：#include<stdio.h>

定义函数：int fflush(FILE \* stream);  
函数说明：fflush()会强迫将缓冲区内的数据写回参数stream指定的文件中，如果参数stream为NULL，fflush()会将所有打开的文件数据更新。  
返回值：成功返回0，失败返回EOF，错误代码存于errno中。  
fflush()也可用于标准输入（stdin）和标准输出（stdout），用来清空标准输入输出缓冲区。  
stdin是standard input的缩写，即标准输入，一般是指键盘；标准输入缓冲区即是用来暂存从键盘输入的内容的缓冲区。stdout是standard output 的缩写，即标准输出，一般是指显示器；标准输出缓冲区即是用来暂存将要显示的内容的缓冲区。

代码一

**[cpp]** [view plain](https://blog.csdn.net/qiqiaiairen/article/details/52317570) [copy](https://blog.csdn.net/qiqiaiairen/article/details/52317570)

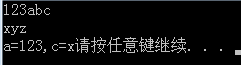
1. #include "stdafx.h"
2. #include<stdlib.h>
3. #include<stdio.h>
5. #pragma warning(disable:4996)
7. **using** **namespace** std;
9. **int** \_tmain(**int** argc, \_TCHAR\* argv[])
10. {
11. **int** a;
12. **char** c;
13. scanf("%d", &a);
14. c = getchar();
16. printf("a=%d,c=%c", a, c);
18. **return** 0;
19. }

https://img-blog.csdn.net/20160825194131775

代码二

**[cpp]** [view plain](https://blog.csdn.net/qiqiaiairen/article/details/52317570) [copy](https://blog.csdn.net/qiqiaiairen/article/details/52317570)

1. #include "stdafx.h"
2. #include<stdlib.h>
3. #include<stdio.h>
5. #pragma warning(disable:4996)
7. **using** **namespace** std;
9. **int** \_tmain(**int** argc, \_TCHAR\* argv[])
10. {
11. **int** a;
12. **char** c;
13. scanf("%d", &a);
14. fflush(stdin);
15. c = getchar();
17. printf("a=%d,c=%c", a, c);
19. **return** 0;
20. }



对比上面的代码，代码一没有清空输入缓冲区，回车时，将123赋值给a，缓冲区剩下abc，接着执行getchar()，发现缓冲区有内容，就无需等待用户输入，直接读取了，将'a'赋给c。代码二执行到fflush()，清空缓冲区，getchar()发现缓冲区没有内容，需要等待用户输入，所以必须输入两次。