

gets和scanf区别

[scanf 和 gets 读取字符串](#)

[深入了解scanf\(\)/getchar\(\)和gets\(\)等函数](#)

[scanf与gets函数读取字符串的区别](#)

今天看到一段话，大致是说gets比scanf()快，有点吃惊，搜了一下，scanf()和gets的区别大致有着几条：

1.scanf() 会忽略行开头的所有空格，并以空格、换行符结束输入；

使用getchar()读取scanf语句执行后，缓冲区留下的换行符，

gets读入以任何字符开始的字符串，以换行符结束，但之后会丢弃换行符并以'\0'代替；

2. [在数据大量的情况下，用gets读取快于scanf\(\)10倍以上](#)（注：来自pudn，只有一段开头话）

3.

第一：要注意不同的函数是否接受空格符、是否舍弃最后的回车符的问题！

读取字符时：

scanf()以Space、Enter、Tab结束一次输入，不会舍弃最后的回车符（即回车符会残留在缓冲区中）；

getchar()以Enter结束输入，也不会舍弃最后的回车符；

读取字符串时：

scanf()以Space、Enter、Tab结束一次输入

gets()以Enter结束输入（空格不结束），接受空格，会舍弃最后的回车符！

第二：为了避免出现上述问题，必须要清空缓冲区的残留数据，可以用以下的方法解决：

方法1：C语言里提供了函数清空缓冲区，只要在读数据之前先清空缓冲区就没问题了！

这个函数是fflush(stdin)。

方法2：自己取出缓冲区里的残留数据。

（说实话这个语句我也没看懂，呵呵！为什么格式控制是这样的！希望高手指点一下！）

```
scanf("%[^\\n]",string);
```

实验代码：单词接龙，输入 n 和n个单词 判断 这些单词是否是首尾相同的



```
#include<iostream>
#include<string.h>
using namespace std;
```

```
int main()
{
```

```

int n;
char word_pre[11];
char word[11];
bool isNo = false;
while (scanf ("%d", &n) != EOF)
{
    //fflush(stdin);
    getchar();
    isNo = false;
    for(int i=0;i<n;i++)
    {
        if(i == 0)
            //scanf("%s",word_pre);
            gets(word_pre);
        else
        {
            //scanf("%s",word);
            gets(word);
            if(word_pre[strlen(word_pre)-1] == word[0])
            {
                strcpy(word_pre,word);
            }
            else
            {
                isNo = true;
                break;
            }
        }
    }
    if(isNo)
        printf("No\n");
    else printf("Yes\n");
}
return 0;
}

```



Language	C	C++	Pascal
To read numbers	int n; while(scanf("%d",	int n; while (cin >> n)	var n: integer; ...

	<pre> &n) != EOF) { ... } </pre>	<pre> { ... } </pre>	<pre> while not seekeof do begin read(n); ... end; </pre>
To read character s	<pre> int c; while ((c = getchar()) != EOF) { ... } </pre>	<pre> char c; while (cin.get(c)) { ... } </pre>	<pre> var c: char; ... while not eof do begin read(c); ... end; </pre>
To read lines	<pre> char line[1024] ; while(get s(line)) { ... } </pre>	<pre> string line; while (getline(ci n, line)) { ... } </pre>	<pre> var line: string; ... while not eof do begin readln(lin e); ... end; </pre>