



字符输入函数：getchar()

【函数功能】

从标准输入 stdin 获取一个字符。

【函数声明】

```
1. int getchar();
```

【说明】

getchar 函数只能接受单个字符，输入多于一个字符时，只接收第一个字符。getchar 函数等待用户输入，直到按回车键才结束，可用于暂停程序的运行，直到输入一个回车键。如果在程序中连续有两个以上 getchar() 函数，应该一次性输入所需字符，最后再按回车键，否则会把回车作为一个字符传给后面的 getchar() 函数。

【程序示例】

```
1. #include<cstdio>
2. #include<iostream>
3. using namespace std;
4. int main()
5. {
6.     char ch = getchar(); //读入字符
7.     cout << ch << endl;
8.     return 0;
9. }
```



字符输出函数：putchar()

【函数功能】

把参数 character 指定的字符写入到标准输出 stdout 中。

【函数声明】

```
1. int putchar ( int character );
```

【程序示例】

```
1. #include<stdio>
2. int main()
3. {
4.     //定义字符变量 c 并赋值'B'
5.     char c = 'B';
6.     //输出该字符
7.     putchar(c);
8.     //用转义字符输出字母'B'
9.     putchar('\x42');
10.    //用 16 进制 ASCII 码值输出字母'B'
11.    putchar(0x42);
12.    //用 10 进制 ASCII 码值输出字母'B'
13.    putchar(66);
14.    return 0;
15. }
```

【运行结果】

BBBB



字符串输入函数：fgets()

【函数功能】

从指定的流 stream 读取一行，并把它存储在 str 所指向的字符串内。当读取(num-1)个字符时，或者读取到换行符时，或者到达文件末尾时，它会停止，具体视情况而定。

【函数声明】

```
1. char * fgets ( char * str, int num, FILE * stream );
```

【参考程序】

```
1. #include<cstdio>
2. #include<iostream>
3. using namespace std;
4. int main()
5. {
6.     char s[100];
7.     fgets(s, 100, stdin);
8.     cout << '[' << s << ' ';
9.     return 0;
10. }
```

【输入样例】

```
I love C++
```

【输出样例】

```
[I love C++
]
```



字符串输入函数：gets()

【函数功能】

从标准输入 stdin 读取一行，并把它存储在 str 所指向的字符串中。当读取到换行符时，或者到达文件末尾时，它会停止，具体视情况而定。

这个函数已经不再被最新的 C++ 编译器支持，请不要再使用。

【函数声明】

```
1. char * gets ( char * str );
```

【参考程序】

```
1. #include<cstdio>
2. #include<iostream>
3. using namespace std;
4. int main()
5. {
6.     char s[100];
7.     gets(s);
8.     cout << '[' << s << ']' ;
9.     return 0;
10. }
```

【输入样例】

```
I love C++
```

【输出样例】

```
[I love C++]
```

【分析】

可以发现，gets 读入的是完整一行，不包括行尾的换行符。



字符串输出函数：puts()

【函数功能】

把字符串 str 写入到指定的流 stream 中，并且添加一个换行字符。

【函数声明】

```
1. int puts ( const char * str );
```

【程序示例】

```
1. #include<cstdio>
2. using namespace std;
3. int main()
4. {
5.     char s[100] = "I love C++";
6.     printf("[");
7.     puts(s);
8.     printf("]");
9.     return 0;
10. }
```

【运行结果】

```
[I love C++
]
```

【分析】

对于已经声明过的字符串 a，下面的语句是等价的。

```
1. printf("%s\n", a);
2. puts(a);
```