

## conio Library

Visual C++ includes a nonstandard library, *conio*, which contains a number of low-level input and output routines. These routines do “direct” console I/O and are specific to Visual C++. These routines work directly with the hardware and are not redirectionable. Other C++ compilers also have a *conio* library, but their routines are named slightly different and may work a bit different from the ones listed below.

To use any of these functions, use “`#include <conio.h>`” above `main()`.

Unfortunately these routines are not always compatible with iostream functions, but they do allow such things as

- Reading a char without requiring the enter key (unbuffered input).
- Reading a char with no screen echo.
- Detecting whether a char is waiting to be read.

### Some of the functions are:

<code>int _getch(void)</code>	Reads a character from the keyboard immediately, without waiting for newline (unbuffered or raw input). No screen echo is provided.
<code>int _getche(void)</code>	Behaves exactly like <code>getch</code> , but it provides screen echo. Note however that this screen echo cannot be redirected.
<code>int _kbhit(void)</code>	Returns non-zero integer (true) if keystrokes are waiting to be read, a 0 otherwise.
<code>int putchar(int Ch)</code>	Displays the character <code>Ch</code> on the console. Return value can be used to check for success, but is usually ignored.
<code>int _cputs(const char * Str)</code>	Displays the string <code>Str</code> on the console.

Example below to count number of characters entered, stops at carriage return.

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

void main()
{
    char Ch;  int Count = 0;
    do
    {
        Ch = _getche();  //get character immediately, non-buffered input
        ++Count;        //count all characters
    }
    while(Ch != '\r');  //'r' is carriage return character

    cout << endl;
    cout << Count << " characters entered";
}

/*****      output      *****/
rd5g3
6 characters entered      */
```