cin.getline() skips

Ok here is a very simple question: In the following code section, the "cin.getline()" doesn't run:

```
cout<< "Specify USB drive letter: ";
char usbD[1];
char outputLoc [40];
cin.getline(usbD, 1, '\n');
cout<< "\n" << usbD << "\n";
```

What am I doing wrong?

c++   getline   cin

edited Sep 21 '12 at 0:03	asked Sep 20 '12 at 22:10
Scooter	Nigh7Sh4de
2,382 2 11 32	**67** 1 13

---

1   Why not just `char usbD; std::cin >> usbD;` ? – chris Sep 20 '12 at 22:16

1   **Always** check the result of read operations once you have attempted to read something. You would have noticed that the `std::cin.getline()` actually *is* run but failed. – Dietmar Kühl Sep 20 '12 at 22:32

---

3 Answers

You need 2 spaces to store a string with a single character, this is because c++ uses a `\0` to delimit the string. You can change your code as follows:

```
cout<< "Specify USB drive letter: ";
char usbD[2];
char outputLoc [40];
cin.getline(usbD, 2, '\n'); // the 2 here will be the drive letter and the ending \0
cout<< "\n" << usbD << "\n";
```

edited Sep 20 '12 at 22:41	answered Sep 20 '12 at 22:26
	Ninten
	153 1 6

---

right i forgot to include space for the terminating character, thanks for your help :) – Nigh7Sh4de Sep 21 '12 at 1:15

You need usbD[2] - for the letter and end of string `'\0'` .

From http://www.cplusplus.com/reference/iostream/istream/getline/

- istream& getline (char* s, streamsize n);
- istream& getline (char* s, streamsize n, char delim);

s

> A pointer to an array of characters **where** the string is stored as a c-string.
>
> n
>
> **Maximum** number of characters to store (including the terminating null character).

edited Sep 20 '12 at 22:25   answered Sep 20 '12 at 22:18

PiotrNycz

10.3k   3   15   50

@PiotrNycz is right, you didn't leave room for the terminating null. But if all you want is one character, there's no reason to use an array.

```
char usbD;
cin.get(usbD);
```

answered Sep 20 '12 at 22:21

Beta

48.2k   5   63   93