

Stack Overflow is a community of 4.7 million programmers, just like you, helping each other. Join them, it only takes a minute:

[Sign up](#)


cin.getline() is skipping an input in C++



If I use the following code, getline doesn't take the last input(for last iteration of "for" loop, it simply skips it) -

```
int main()
{
    int n;
    map<string, set<string> > lst;
    string c,s,c2;

    cin>>n;

    for(int i=0;i<n;i++)
    {
        getline(cin,c); // here it skips the input for last iteration
        stringstream ss;
        ss<<c;

        bool f=1;
        while(ss>>s)
        {
            if(f)
            {
                c2=s;
                f=0;
            }
            else
                lst[c2].insert(s);
        }
    }

    for (map<string, set<string> >::const_iterator ci = lst.begin(); ci != lst.end();
++ci)
    {
        cout<< (*ci).first << " " << (*ci).second.size() <<endl;
    }
}
```

To get rid of it, I put cin.ignore() after getline. Now its taking all the inputs but I'm facing a new issue -

```
#include<iostream>
#include<string>
#include<map>
#include<set>
#include<sstream>
#include<algorithm>

using namespace std;

int main()
{
    int n;
    map<string, set<string> > lst;
    string c,s,c2;

    cin>>n;

    for(int i=0;i<n;i++)
    {
        getline(cin,c);
        cin.ignore();
        stringstream ss;
        ss<<c;

        bool f=1;
        while(ss>>s)
        {
            if(f)
            {
                c2=s;
                f=0;
            }
        }
    }
}
```

```

        else
            lst[c2].insert(s);
    }

    for (map<string, set<string> >::const_iterator ci = lst.begin(); ci != lst.end();
        ++ci)
    {
        cout<< (*ci).first <<" "<< (*ci).second.size() <<endl;
    }
}

```

The new issue is that while taking c2, first character of string gets removed. For example, if I give "England Jane Doe" as input to getline, in c2 I'll get "ngland".

How to get rid of this issue now?

c++ string getline cin

asked Apr 17 '12 at 22:07



theharshest

2,587 5 19 37

2 Mixing >> and getline is delicate. In any case, you never *check* the result of either, so why would you even expect your code to work? – Kerrek SB Apr 17 '12 at 22:10

1 (*ci).first easier to read as ci->first – Loki Astari Apr 17 '12 at 22:15

possible duplicate of using getline(cin, s) after cin – alexisdm Apr 17 '12 at 23:09

4 Answers

This:

```
cin>>n;
```

Is reading the number only.

It leaves the trailing '\n' on the stream.

So your first call to getline() is reading an empty line containing just a '\n'

It is best not to mix the use of operator>> and std::getline(). You have to be very careful on whether you have left the newline on the stream. I find it easiest to always read a line at a time from a user. Then parse the line separately.

```

std::string number;
std::getline(std::cin, number);

int n;
std::stringstream numberline(number);
numberline >> n;

```

edited Jan 29 at 10:16



phadej

5,317 16 38

answered Apr 17 '12 at 22:17



Loki Astari

139k 38 198 363

Thanks for the nice explanation! :) – theharshest Apr 18 '12 at 8:53

Microsoft Azure

Got 5 minutes? Then you have time to deploy your first cloud solution.

Try Azure—free

your cin.ignore() is in the wrong place. getline does not leave the trailing \n newline character, it's the >> symbol which does that.

What you probably want is

```

cin>>n;
cin.ignore();

```

answered Apr 17 '12 at 22:21

Ben Cottrell
2,169 6 17

1 `cin.ignore` only ignores 1 character, you need to ignore all characters until the `\n` newline character. – alexisdm Apr 17 '12 at 23:04

just write

```
cin.ignore();
```

after your last ">>" operator. ">>" operator leaves a newline "\n" in the input bitstream which needs to be cleared before taking the inputs.

answered Jun 22 '14 at 6:23

Vikas Kalwani
18 6

actually the cin buffer has `\n` remaining from last cin, so getline take this `\n` character and gets terminated; ignore the content of buffer before getline...

```
cin.ignore();
```

```
getting(cin, string_name);
```

```
//it will work smoothly
```

edited Nov 22 '14 at 12:30

answered Nov 2 '14 at 11:32

Rajeev Ranjan
11 1

1 This does not provide an answer to the question. To critique or request clarification from an author, leave a comment below their post - you can always comment on your own posts, and once you have sufficient reputation you will be able to comment on any post. – Burimi Nov 2 '14 at 11:55
