

# How to read unknown number of inputs?

Asked 2 years, 5 months ago   Active 1 year, 2 months ago   Viewed 7k times

I am learning C++ using the book C++ Primer.

2 In *Section 1.4.3*, the following example code about reading the unknown number of inputs is given.

```
#include <iostream>
int main()
{
    int sum = 0, value = 0;
    // read until end-of-file, calculating a running total of all values read
    while (std::cin >> value)
        sum += value; // equivalent to sum = sum + value
    std::cout << "Sum is: " << sum << std::endl;
    return 0;
}
```

According to the book, if we give an input of `3 4 5 6`, the output will be `Sum is: 18`

But when I try this on my computer(*Windows 10 with MinGW*), The code does not end. It just keeps on asking for input even if I enter a newline. It works only when I enter a non-int input like `f`.

**Is this expected? If yes, is there any code that stops after inputting a newline?**

I am quite new to c++ and I have already learned python, so getting stuck so early on is quite frustrating.

Thanks and regards.

c++   algorithm   while-loop   c++14   user-input

edited Aug 26 '19 at 11:18



JeJo

20.3k ● 5 ● 27 ● 67

asked May 30 '18 at 15:13



Agile\_Eagle

1,331 ● 1 ● 9 ● 26

Under Linux you would use the EOF character <Ctrl-D> – [kiloalphaindia](#) May 30 '18 at 15:22 ✎

If you want to explicitly read a **line**, you should try `std::getline()`. – [DevSolar](#) May 30 '18 at 15:23 ✎

Either read an entire line (via `std::getline()`) or use program arguments – [Vivick](#) May 30 '18 at 15:29

Is my question wrong, if not why did I get a down vote ? – [Agile\\_Eagle](#) Aug 13 '18 at 11:39 ✎

## 3 Answers

Active Oldest Votes



6



You need to terminate your input by an End-Of-File-character (i.e. `CTRL-Z` on Windows, `CTRL-D` on Mac/Unix), not just by an End-Of-Line (i.e. `Enter`).

A simple `Enter` is interpreted as white space, which will be simply ignored by `operator>>` when reading into an integral data type.

`CTRL-Z` / End-Of-File, in contrast, makes any `operator>>` fail with an error.

See also [this SO answer](#).

Note: Entering `f` will also terminate your loop, since `f` is not considered a valid integral number; Hence, `std::cin >> value` with `value` being of type `int` and an input like `f` will fail as well. To be more accurate: `operator>>` actually returns a reference to the input stream, but if reading in a value fails, `failbit` is set on the stream, and then interpreting the stream object in a boolean expression (implicitly calling `basic_istream::operator bool()`) returns `false`; So maybe the author of the book did not want to explain these details at the respective section in the book :-)

edited May 30 '18 at 16:20



Remy Lebeau

424k ● 26 ● 336 ● 587

answered May 30 '18 at 15:18



Stephan Lechner

33.2k ● 4 ● 27 ● 49

So Shouldn't this be written in the book – Agile\_Eagle May 30 '18 at 15:26

If the section is about explaining `operator >>` and its behaviour, then it should be explained, I suppose. If the example is about demonstrating something else, well... – Stephan Lechner May 30 '18 at 15:33

One last thing, is it possible to write a code which stops after a white space – Agile\_Eagle May 30 '18 at 15:40

Thanks for your patience – Agile\_Eagle May 30 '18 at 15:40



2



Is this expected?

**Yes**, That's what `while (std::cin >> value)` does. See this answer for more explanations: [How is "std::cin>>value" evaluated in a while loop?](#)

is there any code that stops after inputting a newline?

**No**, as `>>` simply ignore a white space(also @StephanLechner mentioned it)

What you can do instead is:

1. Just give a condition; if it satisfies just `break` the loop. you can also provide a console out to make it more understandable to the user. For example:

```
std::cout << "Enter value to sum or press -1 to exit" << std::endl;

while (std::cin >> value && value != -1) // if value == -1, loop stops.
{
    sum += value;
}
```

2. You can simply terminate by the end of character:

- `CTRL-Z` on Windows
- `CTRL-D` on Mac/Unix

edited Jun 20 at 9:12



Community ♦

1 ● 1

answered May 30 '18 at 15:23



JeJo

20.3k ● 5 ● 27 ● 67

1 Will this loop break only if input is -1 – Agile\_Eagle May 30 '18 at 15:27

no, You can also chose the other options here, using the end of file chars, which Stephan Lechner mentioned. – JeJo May 30 '18 at 15:29 ✎

THank you very much – Agile\_Eagle May 30 '18 at 15:40



Is this expected?

1



Yes, as `operator>>` ignores leading whitespace by default, which includes line breaks.



If yes, is there any code that stops after inputting a newline?

Use `std::cin.getline()` or `std::getline()` instead of `operator>>`. You can then use `operator>>` with a `std::istringstream` to parse values from each line, eg:

```
#include <iostream>
#include <string>
#include <sstream>

int main()
{
    std::string line;
    int sum, value;

    do
    {
        std::cout << "Enter a set of numbers, or a blank line to exit: ";
        if (!std::getline(std::cin, line) || line.empty())
            break;
    }
```

```
// read until end-of-line, calculating a running total of all values read
std::istringstream iss(line);
sum = 0;
while (iss >> value)
    sum += value; // equivalent to sum = sum + value

std::cout << "Sum is: " << sum << std::endl;
}
while (true);

return 0;
}
```

[Live Demo](#)

answered May 30 '18 at 16:30



**Remy Lebeau**

424k ● 26 ● 336 ● 587