

In this challenge, we work with *string streams*.

stringstream is a stream class to operate on strings. It implements input/output operations on memory (string) based streams. *stringstream* can be helpful in different type of parsing. The following operators/functions are commonly used here

- *Operator >>* Extracts formatted data.
- *Operator <<* Inserts formatted data.
- *Method str()* Gets the contents of underlying string device object.
- *Method str(string)* Sets the contents of underlying string device object.

Its header file is *sstream*.

One common use of this class is to parse comma-separated integers from a string (e.g., "23,4,56").

```
stringstream ss("23,4,56");
char ch;
int a, b, c;
ss >> a >> ch >> b >> ch >> c; // a = 23, b = 4, c = 56
```

Here *ch* is a storage area for the discarded commas.

If the `>>` operator returns a value, that is a true value for a conditional. Failure to return a value is false.

Given a string of comma delimited integers, return a vector of integers.

Function Description

Complete the *parseInts* function in the editor below.

parseInts has the following parameters:

- *string str*: a string of comma separated integers

Returns

- *vector<int>*: a vector of the parsed integers.

Note You can learn to push elements onto a vector by solving the first problem in the STL chapter.

Input Format

There is one line of *n* integers separated by commas.

Constraints

The length of *str* is less than 8×10^5 .

Sample Input

23, 4, 56

Sample Output

23
4
56