Reverse a List

You are given a list of N elements. Reverse the list without using the *reverse* function. The input and output portions will be handled automatically. You need to write a function with the recommended method signature.

Input Format

Each element, X, of the list is displayed on a separate line.

Output Format

The output is the reverse of the input list.

Sample Input

```
19
22
3
28
26
17
18
4
28
0
```

Sample Output

```
0
28
4
18
17
26
28
3
22
19
```

Method Signature

```
Number Of Parameters: 1
Parameters: [list]
Returns: List or Vector
```

Constraints

$1 <= N <= 100$
$0 <= X <= 100$.

For Hackers Using Clojure

This will be the outline of your function body (fill in the blank portion marked by underscores):

```
(fn[lst]_____________________________)
```

For Hackers Using Scala

This will be the outline of your function body (fill in the blank portion marked by underscores):

```
def f(arr:List[Int]):List[Int] = ___________________
```

For Hackers Using Haskell

This will be the outline of your function body (fill in the blank portion marked by underscores):

```
rev l = ___________________
```

For Hackers Using other Languages

You have to read input from standard input and write output to standard output. Please follow the input/output format mentioned above.

NOTE: You only need to submit the code above after filling in the blanks appropriately. The input and output section will be handled by us. The focus is on writing the correct function.

Contest ends in 39 minutes

Submissions: 12
Max Score: 10
Difficulty: Easy

Rate This Challenge:

More

Scala

```
2 ▾import scala.io.StdIn.{readLine, readInt}
3
4 ▾def f(arr:List[Int]):List[Int] = arr.reverse //arr is variable input
```

Line: 4 Col: 69

Upload Code as File Test against custom input Run Code Submit Code

Testcase 0 ✔

Congratulations, you passed the sample test case.

Click the **Submit Code** button to run your code against all the test cases.

Compile Time

Compile Message

```
warning: 1 deprecation (since 2.13.3); re-run with -deprecation for details
```

Run Time

Input (stdin)

```
19
22
3
28
26
17
18
4
28
0
```

Your Output (stdout)

```
0
28
4
18
17
26
28
3
22
19
```

Expected Output

```
0
28
4
18
17
26
28
3
22
19
```