

Filter Positions in a List

For a given list with N integers, return a new list removing the elements at odd positions. The input and output portions will be handled automatically. You need to write a function with the recommended method signature.

Input Format

N integers contained in the list, each on a separate line.

Output Format

Output the list with the integers at odd positions removed i.e. the first element, the third element and so on. The relative positions of the remaining even-position elements should be the same as they were in the original array. Each integer will be on a separate line.

NOTE: By *odd positions*, we mean the first, third, fifth, etc position of the array needs to be filtered out. As per programming language conventions, these might (and they often do) correspond to indices **0, 2, 4** and so on.

Sample Input

```
2
5
3
4
6
7
9
8
```

Sample Output

```
5
4
7
8
```

Recommended Method Signature

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

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):

```
f lst = _____
```

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.