Given an array of integers, reverse the given array in place using an index and loop rather than a built-in function.

Example

arr = [1, 3, 2, 4, 5]

Return the array [5, 4, 2, 3, 1] which is the reverse of the input array.

Function Description

Complete the function reverseArray in the editor below.

reverseArray has the following parameter(s):

int arr[n]: an array of integers

Return

int[n]: the array in reverse order

Constraints

$1 \le n \le 100$

$0 < arr[i] \le 100$

Input Format For Custom Testing

The first line contains an integer, n, the number of elements in arr.

Each line i of the n subsequent lines (where $0 \le i < n$) contains an integer, arr[i].

```
int* reverseArray(int arr_count, int *arr, int *result_count) {
    start =0
    end =len(arr)-1
    while start< end;
    start +=1
    end-=1
    return arr

}
```

yntax Error(s)

```
_tester__.c: In function 'reverseArray':
_tester__.c:40:5: error: 'start' undeclared (first use in this function)
  40 |     start =0
     |     ^~~~~
_tester__.c:40:5: note: each undeclared identifier is reported only once for e
_tester__.c:40:13: error: expected ';' before 'end'
  40 |     start =0
```

An automated cutting machine is used to cut rods into segments. The cutting machine can only hold a rod of minlength or more, and it can only make one cut at a time. Given the array lengths[] representing the desired lengths of each segment, determine if it is possible to make the necessary cuts using this machine. The rod is marked into lengths already, in the order given.

minlength = 7

The rod is initially sum(lengths) = 4 + 3 + 2 = 9 units long. First cut off the segment of length 4 + 3 = 7 leaving a rod 9 - 7 = 2. Then check that the length 7 rod can be cut into segments of lengths 4 and 3. Since 7 is greater than or equal to minlength = 7, the final cut can be made. Return "Possible".

Example

n = 3

```
char* cutThemAll(int lengths_count, long *lengths, long minLength) {
    lengths I[4,3,2,5]
    rod_length =20
    minlength =3



}
```

Syntax Error(s)

```
__tester__.c: In function 'cutThemAll':
__tester__.c:34:5: error: statement with no effect [-Werror=unused-value]
   34 |      lengths I[4,3,2,5]
      |      ^~~~~~~
__tester__.c:34:12: error: expected ';' before 'I'
   34 |      lengths I[4,3,2,5]
      |              ^~
      |              ;
__tester__.c:39:1: error: control reaches end of non-void function [-Werror=ret
   39 | }
```