

# Problem 1441: Build an Array With Stack Operations

## Problem Information

**Difficulty:** Medium

**Acceptance Rate:** 80.60%

**Paid Only:** No

**Tags:** Array, Stack, Simulation

## Problem Description

You are given an integer array `target` and an integer `n`.

You have an empty stack with the two following operations:

\* \*\*\* "Push": pushes an integer to the top of the stack. \* \*\*\* "Pop": removes the integer on the top of the stack.

You also have a stream of the integers in the range `[1, n]`.

Use the two stack operations to make the numbers in the stack (from the bottom to the top) equal to `target`. You should follow the following rules:

\* If the stream of the integers is not empty, pick the next integer from the stream and push it to the top of the stack. \* If the stack is not empty, pop the integer at the top of the stack. \* If, at any moment, the elements in the stack (from the bottom to the top) are equal to `target`, do not read new integers from the stream and do not do more operations on the stack.

Return \_the stack operations needed to build\_ `target` following the mentioned rules. If there are multiple valid answers, return \*\*any of them\*\*.

\*\*Example 1:\*\*

\*\*Input:\*\* target = [1,3], n = 3 \*\*Output:\*\* ["Push","Push","Pop","Push"] \*\*Explanation:\*\*

Initially the stack s is empty. The last element is the top of the stack. Read 1 from the stream and push it to the stack. s = [1]. Read 2 from the stream and push it to the stack. s = [1,2]. Pop

the integer on the top of the stack.  $s = [1]$ . Read 3 from the stream and push it to the stack.  $s = [1,3]$ .

**Example 2:**

**Input:** target = [1,2,3], n = 3 **Output:** ["Push", "Push", "Push"] **Explanation:** Initially the stack  $s$  is empty. The last element is the top of the stack. Read 1 from the stream and push it to the stack.  $s = [1]$ . Read 2 from the stream and push it to the stack.  $s = [1,2]$ . Read 3 from the stream and push it to the stack.  $s = [1,2,3]$ .

**Example 3:**

**Input:** target = [1,2], n = 4 **Output:** ["Push", "Push"] **Explanation:** Initially the stack  $s$  is empty. The last element is the top of the stack. Read 1 from the stream and push it to the stack.  $s = [1]$ . Read 2 from the stream and push it to the stack.  $s = [1,2]$ . Since the stack (from the bottom to the top) is equal to target, we stop the stack operations. The answers that read integer 3 from the stream are not accepted.

**Constraints:**

$* \ 1 \leq \text{target.length} \leq 100 \ * \ 1 \leq n \leq 100 \ * \ 1 \leq \text{target}[i] \leq n \ * \ \text{target}$  is strictly increasing.

## Code Snippets

**C++:**

```
class Solution {
public:
vector<string> buildArray(vector<int>& target, int n) {
    }
};
```

**Java:**

```
class Solution {
public List<String> buildArray(int[] target, int n) {
    }
}
```

```
}
```

### Python3:

```
class Solution:  
    def buildArray(self, target: List[int], n: int) -> List[str]:
```