59. Build an Array With Stack Operations

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"]

Example 2:

Input: target = [1,2,3], n = 3

Output: ["Push","Push","Push"]

AIM: To build an array with stack

PROGRAM:

```
def buildArray(target, n):
    result = []
    target_index = 0
    for num in range(1, n + 1):
        if target_index == len(target):
            break
        result.append("Push")
        if target[target_index] != num:
            result.append("Pop")
```

```
        else:
            target_index += 1
    return result
target1 = [1, 3]
n1 = 3
print(buildArray(target1, n1))
target2 = [1, 2, 3]
n2 = 3
print(buildArray(target2, n2))
```

OUTPUT:

```
['Push', 'Push', 'Pop', 'Push']
['Push', 'Push', 'Push']
```

TIME COMPLEXITY: O(n)