# 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"]

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].

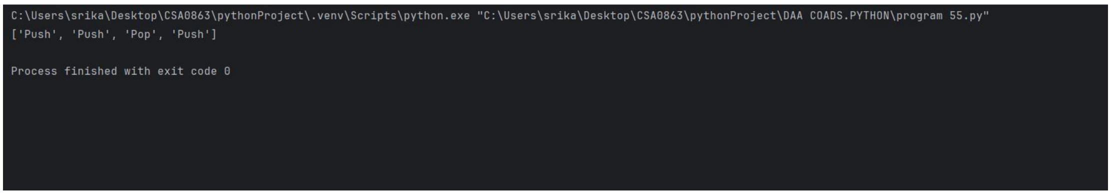
Pop the integer on the top of the stack. s = [1].

Read 3 from the stream and push it to the stack. s = [1,3].

Program:

```
def buildArray(target, n):
    operations = []
    target_index = 0
    for num in range(1, n + 1):
        if target_index < len(target) and num ==
target[target_index]:
            operations.append("Push")
            target_index += 1
        else:
            operations.append("Push")
            operations.append("Pop")
        if target_index == len(target):
            break
    return operations
target = [1, 3]
n = 3
print(buildArray(target, n))
```

Output:



```
C:\Users\srika\Desktop\CSA0863\pythonProject\.venv\Scripts\python.exe "C:\Users\srika\Desktop\CSA0863\pythonProject\DAA COADS.PYTHON\program 55.py"
['Push', 'Push', 'Pop', 'Push']

Process finished with exit code 0
```

Time complexity:

O(n)