A stack is a data structure that has some parallels with an array. Both have methods that include push() and pop(). A stack, however, has the special LIFO property (Last In First Out). Think of it as a stack of dishes. To take dishes off of the stack, you grab the one on top and remove it.

The following are common stack operations:

**push**: pushes a new element on top of a stack;

**pop**: removes the top element from a stack;

**peek**: takes the top element from a stack, but, unlike pop, doesn't remove the element;

**swap**: swaps the position of the top two elements.

The given script performs these stack operations. The function is called **stack()** and contains two parameters:

**stackOperation**: this parameter prescribes the stack operation, such as push and pop.

**stackValue**: this parameter prescribes the item to be pushed on top of a stack. This parameter is only useful when we want to push an item on the stack.

Our stack already contains some data:

storage : [

1,

'{id: 1,value: "obj"}',

"stringHolder",

46

]

The stack function should return an array (even if there's just one position to return).

Our operations are supposed to manipulate this stack based on the **stackOperation** parameter. However, the initial code doesn't function as described above. **Your task is to investigate the code and fix the bugs.**