

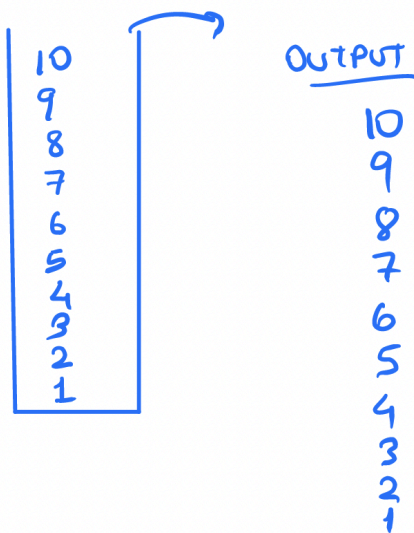
- ```

ArrayStack<int> s;
int a = 22, b = 44;
s.push(2);
s.push(a);
s.push(a + b);
b = s.top();
s.pop();
s.push(b);
s.push(a - b);
s.pop();
while (!s.empty()) {
 cout << s.top() << endl;
 s.pop();
}

```

6. What output is displayed after the following segment of code executes:

```
ArrayStack <int> s;
for (int i = 1; i <= 10; i++)
 s.push(i);
while (!s.empty()){
 cout << s.top() << endl;
 s.pop();
}
```



7.

Suppose you have a stack in which the values 1 through 5 must be pushed on the stack in that order, but that an item on the stack can be popped at any time. Give a sequence of push and pop operations such that the values are popped in the following order:

a. 2, 4, 5, 3, 1

b. 1, 3, 5, 4, 2

a)

```
push(1)
push(2)
pop() → 2
push(3)
push(4)
pop() → 4
push(5)
pop() → 5
pop() → 3
pop() → 1
```

b)

```
push(1)
pop() → 1
push(2)
push(3)
pop() → 3
push(4)
push(5)
pop() → 5
pop() → 4
pop() → 2
```

8. Suppose you have three stacks s1, s2, s2 with starting configuration shown on the left, and finishing condition shown on the right. Give a sequence of push and pop operations that take you from start to finish.

| <i>start</i> |     |     |
|--------------|-----|-----|
| A            |     |     |
| B            |     |     |
| C            |     |     |
| D            |     |     |
| ---          | --- | --- |
| s1           | s2  | s3  |

```

s2.push(s1.top());
s1.pop();
s2.push(s1.top());
s1.pop();
s2.push(s1.top());
s1.pop();
s2.push(s1.top());
s1.pop();
s3.push(s2.top());
s2.pop();
s3.push(s2.top());
s2.pop();
s3.push(s2.top());
s2.pop();
s3.push(s2.top());
s2.pop();

```

| <i>finish</i> |    |     |
|---------------|----|-----|
|               |    | A   |
|               |    | B   |
|               |    | C   |
|               |    | D   |
|               |    | --- |
| s1            | s2 | s3  |

9. Same question in 8, but now suppose the finish configuration on s3 is BDAC (with B on top) ?

```

s2.push(s1.top());
s1.pop();
s2.push(s1.top());
s1.pop();
s3.push(s1.top());
s1.pop();
s1.push(s2.top());
s2.pop();
s3.push(s2.top());
s2.pop();
s2.push(s1.top());
s1.pop();
s3.push(s1.top());
s1.pop();
s3.push(s2.top());
s2.pop();

```