

# Lab 4: Abstract Data Type Solution

## 1) Stack review

1. Write a function that takes a stack of integers and removes all of the items which are greater than 5. The other items in the stack, and their relative order, should remain unchanged.

```
1  ...
2  def remove_big(s: Stack) -> None:
3      """Remove the items in <stack> that are greater than 5.
4
5      Do not change the relative order of the other items.
6
7      >>> s = Stack()
8      >>> s.push(1)
9      >>> s.push(29)
10     >>> s.push(8)
11     >>> s.push(4)
12     >>> remove_big(s)
13     >>> s.pop()
14     4
15     >>> s.pop()
16     1
17     >>> s.is_empty()
18     True
19     """
20     side_stack = Stack()
21
22     # 1. Move all elements in stack 1 to stack 2
23     while not s.is_empty():
24         # 1.1 While moving elements, if element value is greater
25         than 5, then pass
26         popped_element = s.pop()
27
28         if type(popped_element) == int and popped_element > 5:
29             continue
30
31         side_stack.push(popped_element)
32
33     # 2. Move back all elements from stack 2 back to stack 1.
34     while not side_stack.is_empty():
35         s.push(side_stack.pop())
```

```
35     ...
36
```

Listing 1: task\_1\_solution.py

2. Write a function that takes a stack and returns a new stack that contains each item in the old stack twice in a row. We'll leave it up to you to decide what order to put the copies into in the new stack.

Note that because the docstring doesn't say that the old stack will be mutated, the old stack should remain unchanged when the function returns.