Question 1. [5 MARKS]

Consider the following code:

```
class BagOfStuff:
    def __init__(self, stuff: List[str]) -> None:
        self.stuff = stuff

if __name__ == '__main__':
    some_stuff = ['wallet', 'phone', 'keys']
    my_stuff = BagOfStuff(some_stuff)
    some_stuff.append('snacks')
    your_stuff = BagOfStuff(some_stuff)
    assert len(my_stuff.stuff) == 3 and len(your_stuff.stuff) == 4
```

When we run this code, the assertion fails!

Part (a) [3 MARKS]

To debug the code, we have begun to trace the code in the `__main__` block. The memory model diagram shows the state of memory right after the line `some_stuff = ['wallet', 'phone', 'keys']` has been executed. Complete the diagram to show the state of memory immediately before the `assert` line runs.

You can trace through the calls on the call stack if it helps you, but you do not have to include them in your final diagram. You don't have to include anything from the `assert` line in your diagram.