CSC148 - Python recap: Tracing functions and parameters

For each snippet of code below:

(i) Complete the diagram to show the state of the program's memory immediately before the helper function (a1, f1, etc.) returns.

(ii) Write what the program would print, or describe the error that occurs when the program is run.

```python
# Q1.
def a1(n: int) -> int:
    n = n + 1
    return n * 2

if __name__ == '__main__':
    n = 100
    print(a1(n))
    print(n)
```

```python
# Q2.
def f1(thing: list) -> None:
    thing = ['x', 1] + thing

if __name__ == '__main__':
    phone = [9]
    f1(phone)
    print(phone)
```

---

```
# Q1.
def a1(n: int) -> int:
    n = n + 1
    return n * 2

if __name__ == '__main__':
    n = 100
    print(a1(n))
    print(n)
```

```
# Q2.
def f1(thing: list) -> None:
    thing = ['x', 1] + thing

if __name__ == '__main__':
    phone = [9]
    f1(phone)
    print(phone)
```
# Q3.
def f2(thing: list) -> None:
    thing.extend(['x', 1])

if __name__ == '__main__':
    phone = [0]
f2(phone)
print(phone)

# Q4.
def f3(thing: tuple, new: object) -> None:
    thing[1] = new

if __name__ == '__main__':
    phone = (4, 1, 6)
f3(phone, 'surprise!!')
print(phone)