In [1]:
```
class R:
    def __init__(self, x):
        self.x = x + 2

    def __repr__(self):
        return f'R({self.x})'

r1 = R(4)
print(r1)
r2 = R(9)
print(r2)
r3 = R(4)
print(r3)

print(r1 == r2)
print(r1 == r3)
```

```
R(6)
R(11)
R(6)
False
False
```

In [2]:
```
class R:
    def __init__(self, value):
        if isinstance(value, R):
            self.x = 10 * value.x
        else:
            self.x = value

    def __repr__(self):
        return f'R({self.x})'

    def __eq__(self, other):
        return (isinstance(other, R) and (abs(self.x - other.x) <= 2))

r1 = R(1)
r2 = R(7)
print(r1 == r2)

r3 = R(r2)
print(r3)
print(r1 == r3)
```

```
False
R(70)
False
```

In [3]:

```
class H:
    m = 0
    def __init__(self, e, f):
        self.g = e + f
        self.n = H.m
        H.m += 1

    def __repr__(self):
        return f'H(g={self.g},n={self.n})'

h1 = H(8, 1)
h2 = H(2, 9)
print(h1)
print(h2)
print(H.m)
```

```
H(g=9,n=0)
H(g=11,n=1)
2
```

In [4]:

```
class A:
    def __init__(self, L):
        self.L = L

    def __repr__(self):
        return f'A(L={self.L})'

    def g(self, other):
        M = other.L + self.L
        return A(M)

a1 = A([3, 5])
a2 = A([1])
a3 = a1.g(a2)
print(a3)
a4 = a2.g(a1)
print(a4)
```

```
A(L=[1, 3, 5])
A(L=[3, 5, 1])
```