1. (7.0 points) What Would Python Display?

Assume the following code has been executed. The `Link` class appears on the midterm 2 study guide (page 2, left side).

```
def shake(it):
    if it is not Link.empty and it.rest is not Link.empty:
        if it.first + 1 < it.rest.first:
            it.rest = Link(it.rest.first-1, it.rest)
            shake(it)
        else:
            shake(it.rest)
it = Link(2, Link(5, Link(7)))
off = Link(1, it.rest)
shake(it)

def cruel(summer):
    while summer is not Link.empty:
        yield summer.first
        summer = summer.rest
        if summer is not Link.empty:
            summer = summer.rest
summer = Link(1, Link(2, Link(3, Link(4))))
```

Write the output printed for each expression below or *Error* if an error occurs.

(a) (2.0 pt) `print(it)`

- ○ `<2 5 7>`
- ○ `<2 4 5 7>`
- ○ `<2 4 5 6 7>`
- ○ `<2 3 4 5 7>`
- ○ `<2 4 3 5 7>`
- ● `<2 3 4 5 6 7>`
- ○ `<2 4 3 5 6 7>`

(b) (2.0 pt) `print(off)`

`<1 5 6 7>`

(c) (2.0 pt) `print([x*x for x in cruel(summer)])`

`[1, 9]`

(d) (1.0 pt) What is the order of growth of the time it takes to evaluate `shake(Link(1, Link(n)))` in terms of `n`?

- ◯ exponential
- ◯ quadratic
- ● linear
- ◯ constant