[[questions]]
id = "bba8893d-f59f-492a-b639-ea04a5c8093b"
type = "Tracing"
prompt.program = """
#[derive(Copy, Drop)]
struct Point {
  x: u32,
  y: u32,
}
fn main() {
  let mut a = Point { x: 1, y: 2 };
  a.x += 1;
  let b = Point { y: 1, ..a };
  a.x += 1;
  println!("{}", b.x);
}
"""
answer.doesCompile = true
answer.stdout = "2"
context = """
The `..a` syntax copies each field of `a` into `b` (except `y`), so the second `a.x += 1` has no effect on `b`.
"""
[[questions]]
id = "77ef68d6-ee7f-4642-8817-73b846c46b2b"
type = "Tracing"
prompt.program = """
#[derive(Copy, Drop)]
struct Point {
    x: u32,
    y: u32,
}
fn main() {
    let mut p = Point { x: 1, y: 2 };
    let mut x = p.x;
    let mut y = p.y;
    x += 1;
    y += 1;
    println!("{} {}", p.x, p.y);
}
"""
answer.doesCompile = true
answer.stdout = "1 2"
context = """
`p.x` and `p.y` are __copied__ into new `x` and `y` variables, so the `x += 1` and `y += 1` lines have no effect on `p`.

''''''