```
principal A
principal B

def cmp : ℤ{isec:A,B} → 𝔹{yao:A,B}
def cmp = λ xy →
  let x : ℤ{yao:A,B}
  let x = share{yao:A,B} xy.A
  let y : ℤ{yao:A,B}
  let y = share{yao:A,B} xy.B
  let r : 𝔹{yao:A,B}
  let r = x ≤ y
  in r

def cmp-rev : 𝟙 →{inp:A,B;rev:A,B} 𝔹{ssec:A,B}
def cmp-rev = λ • →
  let r : 𝔹{yao:A,B}
  let r = cmp-mpc •
  let p : 𝔹{ssec:A,B}
  let p = reveal{A,B} r
  in p

def one-liner : 𝟙 →{inp:A,B;rev:A,B} 𝔹{ssec:A,B}
def one-liner = λ • →
  let xy = {par:A,B} read ℤ "e1-input.txt"
  in reveal{A,B} (share{yao:A,B} xy.A) ≤ (share{yao:A,B} xy.B)

def main : 𝔹{ssec:A,B} × 𝔹{ssec:A,B}
def main = cmp-mpc-rev • , one-liner •
```