```
_let f = _fun (x)
           _fun (g)
             g(x + 1)
_in _let g = _fun (y)
              y + 2
_in (f(5))(g)
```

A. 6

B. 7

C. 8

D. error

```
_let f = _fun (x)
           _fun (g)
             g(x + 1)
_in _let g = _fun (y)
              y + 2
_in (f(5))(g)
```

A. 6

B. 7

[✓ **C.** 8]

D. error

Interp Sketch

```
 _let f = _fun (x)
           _fun (g)
              g(x + 1)
_in _let g = _fun (y)
               y + 2
_in (f(5))(g)

_let g = _fun (y)
               y + 2
_in ((_fun (x)
        _fun (g)
           g(x + 1))(5))(g)


((_fun (x)
    _fun (g)
      g(x + 1))(5))(_fun (y)
                      y + 2)

(_fun (g)
  g(5 + 1))(_fun (y)
             y + 2)

(_fun (y)
   y + 2)(5 + 1)

(_fun (y)
   y + 2)(6)

6 + 2
```

Summary

calling a function uses substitution for the argument

a function is a value, just like a number or boolean

a function isn't required to have a name

`_fun` **`(`**⟨variable⟩**`)`** ⟨expr⟩ can appear anywhere

a function can take a function argument

a function can return a function