

Section 09, Problem 1 (Part 1)

Consider the following OCaml code:

```
let x = ref 0 ;;
let y = ref 0 ;;

let _ =
  (y := !y + 1; if 0 < !x then x else y)
  :=
  (x := !x + 1; if 0 < !y then !x else !y) ;;

x, y ;;
```

When run in utop it produces the following output:

```
val x : int ref = {contents = 0}
val y : int ref = {contents = 0}
- : unit = ()
- : int ref * int ref = ({contents = 0}, {contents = 1})
```

Explain why.

Section 09, Problem 1 (Part 2)

Now consider this OCaml code:

```
let a = ref 0
let b = ref 1
let c = ref 2 ;;

(if (a := !b + !c; !b < !c) then
```

```
(a := !a * 2; c) else
(c := !c + 1; b := !b + !c; b)) := !c * 2 ;;
```

```
a, b, c ;;
```

When run in utop, what will it produce?

A	({contents = 6}, {contents = 1}, {contents = 4})
B	({contents = 7}, {contents = 4}, {contents = 6})
C	({contents = 6}, {contents = 8}, {contents = 3})
D	({contents = 6}, {contents = 4}, {contents = 6})
E	None of the above

Section 09, Problem 1 (Part 3)

Now consider this OCaml code:

```
let x = ref 2
let y = ref 3
```

```
let f x = (x := !x + 1; !y) + (y := !x + !y; !x * 2) ;;
```

```
let res = f (x := !y + 1; y) ;;
```

```
res, x, y ;;
```

When run in utop, what will it produce?

A	(9, {contents = 6}, {contents = 5})
B	(13, {contents = 5}, {contents = 8})
C	(20, {contents = 4}, {contents = 8})
D	(19, {contents = 4}, {contents = 6})
E	None of the above