

Exercise 1

a

Det ville printe x: 3 og y: 1, fordi de ikke bliver ændret når man bruger call by value.

b

Den printer flg. fordi a og c peger til samme adresse. r: 19 x: 8, y: 3

c

Den ville printe flg. fordi at y og x bliver ændret i funktionen. x bliver ændret 2 gange, men den sidste reference er ved c. r: 13 x: 7 y: 2

Exercise 2

a

f(3) ville printe x: 4, da den tjekker om den lokale x er tre, hvilket det er og kalder en funktion g som bare printer den globale x.

f(5) ville printe 4 af samme grund.

b

f(3) ville printe x: 3 da den ville overskrive den første erklærede instans af x, hvilket er den globale. Derefter er x = 3 hvilket forårsager at g bliver kaldt og den printer så x: 3

f(5) ville printe 7, fordi den globale x altid bliver overskrevet af nye erklæringer.

exercise 3

a

fun char main() = let x = read(char) in if x == 'c' then x else 1

I eksemplet ovenfor vil interpreteren ikke opdage fejlen før if statementet og derfor vil den først printe en fejl når x != 'c'.

b

Scan:

$\forall a. \forall b. ((a * b) \rightarrow b * [a]) \rightarrow [b]$

1. Udregn typen af et element fra $a : t$
2. udregn typen af expr $e : te$
3. Tjek om typerne matcher i $f table$
4. IF $argc$ (arg count) != 2 THEN error ELSE good
5. IF $typeof(argv[1]) == t$ AND $typeof(argv[2]) == te$ THEN good ELSE error
6. IF $typeof(retval) == t$ THEN good ELSE error

Filter:

$\forall a(a \rightarrow b * [a]) \rightarrow [a], b = bool$

1. Udregn typen af et element fra $a : t$
2. Tjek om typen matcher i $f\text{table}$
3. IF $argc$ (arg count) $\neq 1$ THEN error ELSE good
4. IF $\text{typeof}(argv[1]) == \text{bool}$ THEN good ELSE error
5. IF $\text{typeof}(retVal) == \text{bool}$ THEN good ELSE error