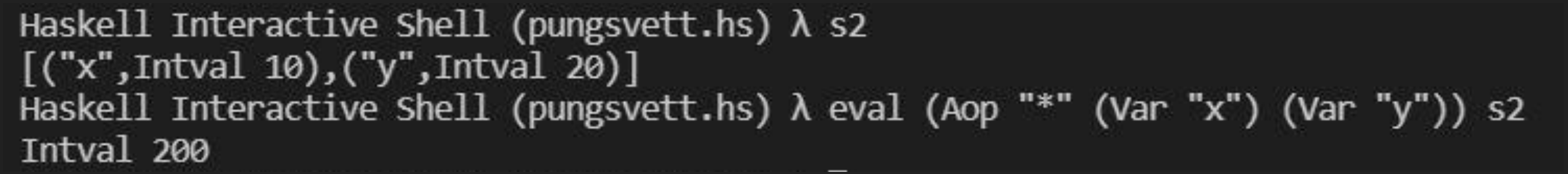
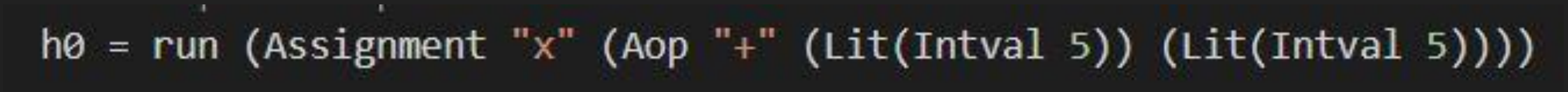
Laboration 3

Johan Sollenius
Henrik Eriksson

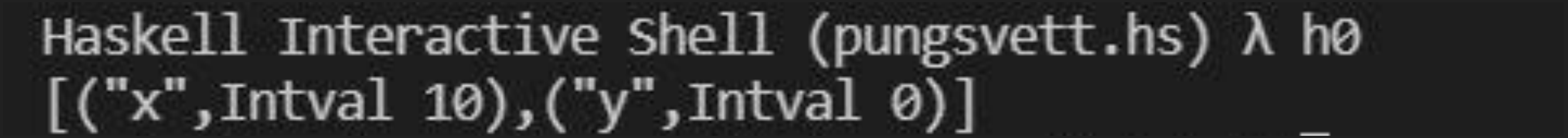
Multiplikation och addition



```
Haskell Interactive Shell (pungsvett.hs) λ s2
[("x",Intval 10),("y",Intval 20)]
Haskell Interactive Shell (pungsvett.hs) λ eval (Aop "*" (Var "x") (Var "y")) s2
Intval 200
```



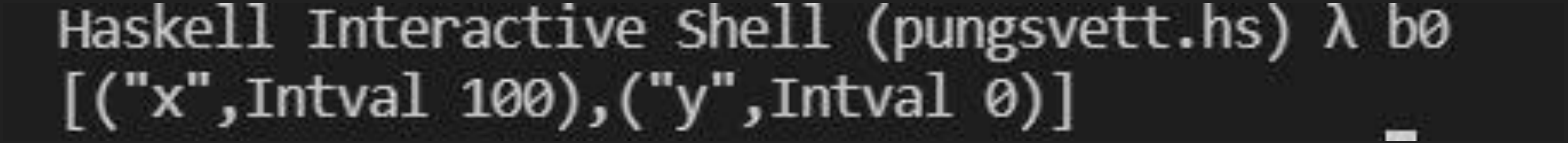
```haskell
h0 = run (Assignment "x" (Aop "+" (Lit(Intval 5)) (Lit(Intval 5))))
```



```
Haskell Interactive Shell (pungsvett.hs) λ h0
[("x",Intval 10),("y",Intval 0)]
```

Enkel if sats utan else (if x = 0, y = 0, then x= 100)

```
b0 = run ( (Conditional (Bop ("&&") (Rop "==" (Var "x") (Lit(Intval 0))) (Rop
"==" (Var "y") (Lit(Intval 0)))) (Assignment "x" (Lit(Intval 100))) Skip))
```



```
Haskell Interactive Shell (pungsvett.hs) λ b0
[("x",Intval 100),("y",Intval 0)]
```

```
Skrivet i Python
if y == 0 && x == 0:
  x = 100
```

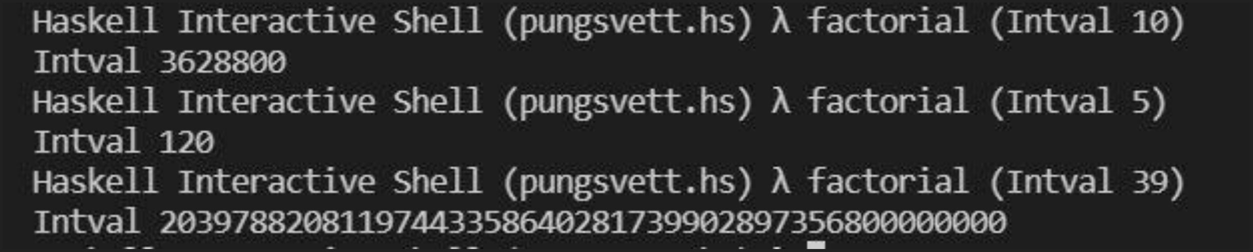
Fakultet

```haskell
c1 :: Statement
c1 = (Assignment  "y" (Var "x"))
-- y = y * (x - 1)
c2 :: Statement
c2 = (Assignment  "y" (Aop "*" (Var "y") (Aop "-" (Var "x") (Lit(Intval  1)))))
-- x = (x - 1)
c3 :: Statement
c3 = (Assignment  "x" (Aop "-" (Var "x") (Lit(Intval  1))))
```

```haskell
factorial  :: Value -> Value
factorial n = get  "y" (run (Block (Nonnil (Assignment  "x" (Lit(n))) (Nonnil c1 (Nonnil (Loop (Rop  "!=" (Var "x") (Lit(Intval  1)))
(Block (Nonnil c2 (Nonnil c3 Nil)))) Nil))))))
```

skrivet i python

```python
def factorial n:
  x = n
  y = x
  while x != 1:
    y = y * (x - 1)
    x = (x - 1)
  return y
```



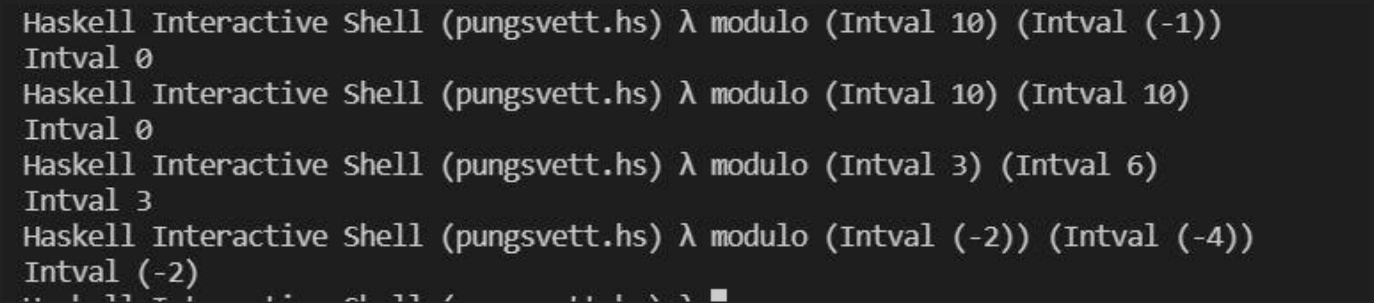
```
Haskell Interactive Shell (pungsvett.hs) λ factorial (Intval 10)
Intval 3628800
Haskell Interactive Shell (pungsvett.hs) λ factorial (Intval 5)
Intval 120
Haskell Interactive Shell (pungsvett.hs) λ factorial (Intval 39)
Intval 20397882081197443358640281739902897356800000000
```

Modulo

```
modulo :: Value -> Value -> Value
modulo a b = get "y" ( run (Block (Nonnil (Assignment "x" (Aop "/" (Lit(a)) (Lit(b)))) (Nonnil
(Assignment "y" (Aop "*" (Lit(b)) (Var "x"))) (Nonnil (Assignment "y" (Aop "-" (Lit(a)) (Var "y")))
Nil)))) )
```

Skrivet i Python

```
def modulo (a,b):
 x = a//b
 y = b * x
 y = a - y
 return y
```



```
Haskell Interactive Shell (pungsvett.hs) λ modulo (Intval 10) (Intval (-1))
Intval 0
Haskell Interactive Shell (pungsvett.hs) λ modulo (Intval 10) (Intval 10)
Intval 0
Haskell Interactive Shell (pungsvett.hs) λ modulo (Intval 3) (Intval 6)
Intval 3
Haskell Interactive Shell (pungsvett.hs) λ modulo (Intval (-2)) (Intval (-4))
Intval (-2)
```