CS3342 – Assignment 3
due Mar. 24, 2022
2-day no-penalty extension until: Mar. 26, 11:55pm
(SRA's cannot be used to extend the due date further)

1. (25pt) Consider the following program:

```
int a;
s(n) { a = n; }
p() { print a; }
f() { s(1); p(); }
g() { int a; s(2); p(); }
s(0); f(); p(); g(); p();
```

a: 1

all are usig the global one.

   (a) (10pt) Give the output of the program if static scoping is used 1 1 2 2

   (b) (15pt) Give the output of the program if dynamic scoping is used. 1 1 2 1

   Run the program step by step to explain the output.

2. (25pt) Consider the Python program:

```
def A(I, P):
    def B():
        print(I)
    if I > 3:
        P()
    elif I > 2:
        A(4, P)
    elif I > 1:
        A(3, B)
    else:
        A(2, B)
def C():
    print(0)
A(1, C)
```

   What does this program print? Explain your answer. Draw the run-time stack. You can use in Python `print(P)` and `print(B)` from within `A` to identify the procedures uniquely.

3. (25pt) We have modelled the non-negative integers and operations with λ-expressions:

$$n \equiv \lambda fc.\, \underbrace{f(f \ldots (f(f}_{nf\text{'s}}\, c)) \ldots)$$
$$+ \equiv \lambda mnab.m\, a\, (n\, a\, b)$$
$$\times \equiv \lambda mna.m\, (n\, a)$$

   Prove, using both call-by-value and call-by-name reduction, that:

   (a) (10pt) $1 + 3 \Longrightarrow_\beta^* 4$

   (b) (15pt) $0 * 2 \Longrightarrow_\beta^* 0$

(a): call-by-value:

$1+3 \equiv +\ 1\ 3$

$\Rightarrow (\lambda mnab.\, ma(nab))(\lambda fc.\, fc)(\lambda fc.\, f(f(fc)))$

$\Rightarrow (\lambda nab.\, (\lambda fc.\, fc)\, a(nab))(\lambda fc.\, f(f(fc)))$

$\Rightarrow (\lambda nab.\, (\lambda c.\, ac)(nab))$ ----

$\Rightarrow (\lambda nab.\, a(nab))(\lambda fc.\, f(f(fc)))$

$\Rightarrow (\lambda ab.\, a((\lambda fc.\, f(f(fc)))ab))$

$\Rightarrow (\lambda ab.\, a((\lambda c.\, a(a(ac)))b))$

$\Rightarrow (\lambda ab.\, a(a(a(ab))))$

....

call-by-name:

$1+3 \equiv +\ 1\ 3$

$\Rightarrow (\lambda mnab.\, ma(nab))(\lambda fc.\, fc)(\lambda fc.\, f(f(fc)))$

$\Rightarrow (\lambda nab.\, (\lambda fc.\, fc)\, a(nab))(\lambda fc.\, f(f(fc)))$

$\Rightarrow (\lambda ab.\, (\lambda fc.\, fc)\, a((\lambda fc.\, f(f(fc)))ab))$

$\Rightarrow (\lambda ab.\, (\lambda c.\, ac)((\lambda fc.\, f(f(fc)))ab))$

$\Rightarrow (\lambda ab.\, a((\lambda fc.\, f(f(fc)))ab))$

$\Rightarrow$ ⋮

$$n \equiv \lambda f c.\, \underbrace{f(f \ldots (f(f}_{n f\text{'s}}\, c)) \ldots)$$
$$+ \equiv \lambda mnab.m\, a\, (n\, a\, b)$$
$$\times \equiv \lambda mna.m\, (n\, a)$$

Prove, using both call-by-value and call-by-name reduction, that:

(a) (10pt) $1+3 \Longrightarrow^*_\beta 4$

(b) (15pt) $0 * 2 \Longrightarrow^*_\beta 0$

(b) $0*2 \equiv *\ 0\ 2.$

call-by-value: $*02$

$\Rightarrow (\lambda mna.\, m(na))(\lambda fc.\, c)(\lambda fc.\, f(fc))$

$\Rightarrow (\lambda na.\, (\lambda fc.\, c)\, \cancel{(na)})(\lambda fc.\, f(fc))$

$\Rightarrow (\lambda na.\, (\lambda c.\, c)\cancel{(na)})(\lambda fc.$ ....

$\Rightarrow (\underline{\lambda n}a.na)\underline{(\lambda zc.z(zc)))}$

$\Rightarrow \lambda a.(\underline{\lambda z}c.z(zc))\underline{a}$

$\Rightarrow \lambda a.(\lambda c.a(ac))$

$\Rightarrow \lambda c.a(ac))$

4. (25pt) We modelled also boolean logic: $\mathtt{T} \equiv \lambda xy.x$, $\mathtt{F} \equiv \lambda xy.y$. We define a new operator, `xor`, that computes the exclusive or of two boolean values, as follows:

$$\mathtt{xor} \equiv \lambda xy.x\,(y\,\mathtt{F}\,\mathtt{T})\,y$$

Prove, using both call-by-value and call-by-name reduction, that:

$$\mathtt{xor}\ \mathtt{T}\ \mathtt{T} \Longrightarrow_{\beta}^{*} \mathtt{F}$$

Q3,4 note: For all computations you perform, indicate clearly the reduction being being done by underlying the abstraction used and the argument it is applied to: $(\underline{\lambda x}.M)\underline{N}$. Do not use anything already computed in the notes; compute everything from scratch.

READ ME! Submit your answers as a *single pdf file* in OWL. Solutions should be typed but readable (by others!) hand-written solutions are acceptable. Source code, if required, is submitted as separate files.

LaTeX: For those interested, the best program for scientific writing is LaTeX. It is far superior to all the other programs, it is free, and you can start using it in minutes; here is an introduction: `https://tobi.oetiker.ch/lshort/lshort.pdf`