CSE340 Spring 2020 HOMEWORK 4
Due by 11:59 PM on Thursday April 2 2020

PLEASE READ THE FOLLOWING CAREFULLY

1. Your answers for questions 1 and 2 must be typed.
2. Your answers to question 3 can be handwritten but it has to be neat and legible. Your answers to question 3 can also be typed if you prefer.
3. On Gradescope, you should submit the answers to separate question separately.
4. For each question, read carefully the required format for the answer. The required format will make it easier for you to answer and for the graders to grade. Answers that are not according to the required format will not be graded

Problem 1 (Lambda Calculus). The goal of this problem is to give you further practice with lambda calculus. Each part of this problem will have an expression that you are asked to evaluate or simplify as much as possible. The following are some examples

Example 1. plus2 = λn. succ (succ n)
what does the following evaluate to: 4 plus2 2
Answer. 10

Example 2. quad = λx. λy. λz. λw. pair (pair x y) (pair z w)
what does the following evaluate to: succ (fst (snd (quad 1 3 5 7)))
Answer. 6

We will use the following definitions in what follows

next1 = λp. pair (times (fst p) (snd p)) (succ (snd p))

next2 = λp. pair ((snd p) (fst p))

next3 = λn. (times n n)

For each of the following, give the value that the expressions evaluates to

1. what is next1 (pair 1 1)? pair(1 2)
2. what is next1 (next1 (pair 1 1))? pair(2 3)
3. what does the function λn. fst (n next1 (pair 1 1)) calculate? Factorial of n
4. what is fst (next2 (pair tru fls))? fls
5. what is fst (next2 (next2 (pair tru fls)))? tru
6. what does the function λn. fst (n next2 (pair tru fls)) calculate? Describe the function in a compact description. Depending on what n is fst(n next2 (pair tru fls)) would be either fls or tru. When n is odd you get fls and when n is even you get tru, So this function would determine if n is even or odd.
7. what is next3 2? 4
8. what is next3 4? 16

9. what does the function λn. n next3 2 calculate? n*(2^2)

Consider the following program written in C syntax

```
int a , b , c ;     // first declaration

void g()
{
   print(a,b,c);
}

int f(int a)        // parameter declaration
{    int b;         // second declaration
     b = a + 1;
     g();           // first call

     { int a;       // third declaration
       int c;       // fourth declaration
       c = b;
       a = b + c;
       g();         // second call
     }

     g();           // third call

     return a + b ;
}

int main()
{
     int a = 2;     // fifth declaration

     a = f(a);
     g();           // fourth call
}
```

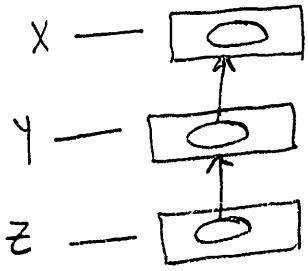
1. If static scoping is used, the reference to a in the first call to g() resolves to which declaration? First
2. If static scoping is used, the reference to a in the third call to g() resolves to which declaration? First
3. If dynamic scoping is used, the reference to a in the first call to g() resolves to which declaration? Fifth
4. If dynamic scoping is used, the reference to a in the third call to g() resolves to which declaration? Fifth
5. What is the output of this program if static scoping is used? 0 0 0
6. What is the output of this program is dynamic scoping is used? 5 3 0

Consider the following C code

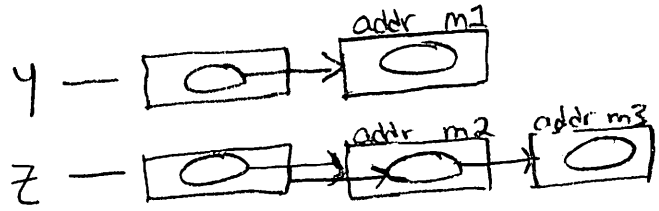
```
int x;                                  // location 1 associated with x
int *y;                                 // location 2 associated with y
int **z;                                // location 3 associated with z

y = &x;                                 // statement 1
z = &y;                                 // statement 2

y = (int *) malloc(sizeof(int));        // statement 3: m1 allocated
z = (int **) malloc(sizeof(int *));     // statement 4: m2 allocated
*z = (int *) malloc(sizeof(int));       // statement 5: m3 allocated

y = *z;                                 // statement 6
```

1. Draw the box circle diagram after statements 1 and 2 are executed
2. Draw the box circle diagram after statements 1 through 5 are executed
3. Draw the box circle diagram after statements 1 through 6 are executed

1)



2)



3)