CS421 : hw3	**Summer 2015**
Dan McQuillan	*Handed In: June 21, 2015*

Problem 1

```
[[ FUN f -> FUN x -> if x > 0 then f x else f ((-1) * x) ]]
        (FUN w -> report w)

= (FUN w -> report w) ( FUN f k0 -> [[ FUN x ->
        if x > 0 then f x else f((-1) * x) ]] k0 )

= (FUN w -> report w) ( FUN f k0 -> k0 ( FUN x k1 ->
        [[ if x > 0 then f x else f((-1) * x) ]] k1 ) )

= (FUN w -> report w) ( FUN f k0 -> k0 ( FUN x k1 ->
        [[ x > 0 ]] ( FUN v ->
                if v then
                        [[ f x ]] k1
                else [[ f( (-1) * x ) ]] k1 ) ) )

= (FUN w -> report w) ( FUN f k0 -> k0 ( FUN x k1 -> (
         FUN v ->
                if v then
                        [[ f x ]] k1
                else [[ f( (-1) * x ) ]] k1 ) ( x > 0 ) ) )

= (FUN w -> report w) ( FUN f k0 ->
    k0 ( FUN x k1 ->
         ( FUN v ->
             if v then
                 [[ f ]] ( FUN e ->
                         [[ x ]] ( FUN v2 -> e v2 k1 ) )
             else [[ f( (-1) * x ) ]] k1 ) ( x > 0 ) ) )

= (FUN w -> report w) ( FUN f k0 ->
    k0 ( FUN x k1 ->
         ( FUN v ->
             if v then
                 ( ( FUN e -> ( ( FUN v2 -> e v2 k1 ) x ) ) f )
             else
                 [[ f( (-1) * x ) ]] k1 ) ( x > 0 ) ) )
```

```
= (FUN w -> report w) ( FUN f k0 ->
    k0 ( FUN x k1 ->
        ( FUN v ->
            if v then
                ( ( FUN e -> ( ( FUN v2 -> e v2 k1 ) x ) ) f )
            else
                [[ f ]] ( FUN e ->
                        [[ (-1) * x ) ]] ( FUN v2 -> e v2 k1 ) ) ) )

= (FUN w -> report w) ( FUN f k0 ->
    k0 ( FUN x k1 ->
        ( FUN v ->
            if v then
                ( ( FUN e -> ( ( FUN v2 -> e v2 k1 ) x ) ) f )
            else
                ( ( FUN e ->
                        [[ (-1) * x ]] ( FUN v2 -> e v2 k1 ) ) f ) ) ) )

= (FUN w -> report w) ( FUN f k0 ->
    k0 ( FUN x k1 ->
        ( FUN v ->
            if v then
                ( ( FUN e -> ( ( FUN v2 -> e v2 k1 ) x ) ) f )
            else
                ( ( FUN e -> [[ -1 ]] ( FUN v3 -> [[ x ]] ( FUN v4 ->
                    ( ( FUN v2 -> e v2 k1 ) ( v3 * v4 ) ) ) ) ) f ) ) ) )

= (FUN w -> report w) ( FUN f k0 ->
    k0 ( FUN x k1 ->
        ( FUN v ->
            if v then
                ( ( FUN e -> ( ( FUN v2 -> e v2 k1 ) x ) ) f )
            else
                ( ( FUN e ->
                    ( ( FUN v3 -> [[ x ]] ( FUN v4 ->
                        ( ( FUN v2 -> e v2 k1 ) ( v3 * v4 ) ) ) ) -1 )
                ) f ) ) ) )

= (FUN w -> report w) ( FUN f k0 ->
    k0 ( FUN x k1 ->
        ( FUN v ->
            if v then
                ( ( FUN e -> ( ( FUN v2 -> e v2 k1 ) x ) ) f )
            else
                ( ( FUN e ->
```

```
                    ( ( FUN v3 ->
                        ( ( FUN v4 ->
                            (
                                ( FUN v2 -> e v2 k1 ) ( v3 * v4 )
                            )
                        ) x )
                    ) -1 )
                ) f )
            )
        )
    )
```

Problem 2

```
type two_three_tree =
| TwoChildren of int * string * (two_three_tree * two_three_tree)
| ThreeChildren of
        int *
        (string * string) *
        (two_three_tree * two_three_tree * two_three_tree)
| LeafOneData of int * string
| LeafTwoData of int * (string * string);;
```

Problem 3

```
type 'a red_black_tree =
| RedNode of 'a red_node
| BlackNode of 'a black_node
and 'a red_node =
| RedTree of 'a * ( 'a black_node * 'a black_node )
and 'a black_node =
| TwoRedNodes of 'a * ( 'a red_node * 'a red_node )
| TwoBlackNodes of 'a * ( 'a black_node * 'a black_node )
| BlackLeaf;;
```