Part 1: Theoretical Questions

1a) False –procedure's g return value is of type T2, which result in F being called with the wrong type.

1b) False – the procedure forces X to be type T2 in contradiction to the variable's statements of the procedure T1.

1c) True – All types are matching.

1d) True – All types are matching.

Question 2:

2a)

((lambda (f x1) (if x1 (f 1 x1) (f 3 x1))) + #t)	T0
(lambda (f x1) (if x1 (f 1 x1) (f 3 x1)))	T1
(if x1 (f 1 x1) (f 3 x1))	T2
x1	Tx
(f 1 x1)	T3
F	Tf
1	Tnum1
(f 3 x1)	T5
3	Tnum3
+	T plus
#t	T true

((lambda (f x1) (if x1 (f 1 x1) (f 3 x1))) + #t)	T1 = [Tplus x Ttrue -> T0]
(lambda (f x1) (if x1 (f 1 x1) (f 3 x1)))	T1 = [Tf x Tx-> T2]
(if x1 (f 1 x1) (f 3 x1))	T2 =[Tx -> T5]
(f 1 x1)	Tf = [Tnum1 x Tx -> T3]
(f 3 x1)	Tf = [Tnum3 x Tx -> T5]

1	Tnum1 = number
3	Tnum3 = number
+	T plus = [Number X number -> Number]
#t	T true = boolean

Equation	Substitution
T1 = [Tplus x Ttrue -> T0]	{ T1 = [Tplus x Ttrue -> T0]}
T1 = [Tf x Tx-> T2]	
T2 =[Tx -> T5]	
Tf = [Tnum1 x Tx -> T3]	
Tf = [Tnum3 x Tx -> T5]	
Tnum1 = number	
Tnum3 = number	
T plus = [Number X number -> Number]	
T true = boolean	

Equation	Substitution
	{ T1 = [T plus x T true -> T0], T2 =[Tx -> T5]}
Tf = [Tnum1 x Tx -> T3]	
Tf = [Tnum3 x Tx -> T5]	
Tnum1 = number	
Tnum3 = number	
T plus = [Number X number -> Number]	
T true = boolean	
Tf = Tplus	
Tx = T true	
T2 = T	

Equation	Substitution
	{ T1 = [T plus x T true -> T0], T2 =[Tx -> T5], Tf = [Tnum1 x Tx -> T3], Tf = [Tnum3 x Tx -> T5]}
Tnum1 = number	
Tnum3 = number	
T plus = [Number X number -> Number]	
T true = boolean	
Tf = [Number X number -> Number]	
Tx = T true	
T2 = T	
Tx = Tx	
T3 = T5	

Equation	Substitution
	{ T1 = [T plus x T true -> T0], T2 =[Tx -> T5], Tf = [number x Tx -> T3], Tf = [number x Tx -> T5], }
T plus = [Number X number -> Number]	
T true = boolean	
Tf = [Number X number -> Number]	
Tx = T true	
T2 = T	
Number = number	
Tx = Tx	
T3 = T5	

Equation	Substitution
	{ T1 = [[Number X number -> Number] x T true -> T0], T2 =[Tx -> T5], Tf = [number x Tx -> T3], Tf = [number x Tx -> T5], }
T true = boolean	
Tf = [Number X number -> Number]	
Tx = T true	
T2 = T	
Number = number	
Tx = Tx	
T3 = T5	

Equation	Substitution
	{ T1 = [[Number X number -> Number] x Boolean -> T0], T2 =[Tx -> T5], [Number X number -> Number] = [number x Boolean -> T3], [Number X number -> Number] = [number x Boolean -> T5], }
Tx = T true	
T2 = T	
Number = number	
Tx = Tx	
T3 = T5	

We got a contradiction with wrong primitive types.

2B)

((lambda (f1 x1 y1) (f1 x1 y1)) * 1 3) -> ((lambda (f x y) (f x y)) * 1 3)

((lambda (f x y) (f x y)) * 1 3)	T0
(lambda (f x y) (f x y))	T1
(f x y)	T2
F	T f
X	Tx
Y	Ty
*	Tmul
1	T num1
3	T num3

((lambda (f x y) (f x y)) * 1 3)	T1 = [Tmul x T num1 x T num3 -> T0]
(lambda (f x y) (f x y))	T1 = [Tf x Tx x Ty-> T2]
(f x y)	Tf =[Tx X Ty -> T2]

1	Tnum1 = number
3	Tnum3 = number
Tmul	T mul = [Number X number -> Number]

Equation	Substitution
T1 = [Tmul x T num1 x T num3 -> T0]	{}
T1 = [Tf x Tx x Ty-> T2]	
Tf =[Tx X Ty -> T2]	
Tnum1 = number	
Tnum3 = number	
T mul = [Number X number -> Number]	

Equation	Substitution
	{ T1 = [Tmul x T num1 x T num3 -> T0] ,T1 = [Tf x Tx x Ty - > T2]
Tf =[Tx X Ty -> T2]	
Tnum1 = number	
Tnum3 = number	
T mul = [Number X number -> Number]	
Tf = Tmul	
Tx = T num1	
Ty = T num 3	

Equation	Substitution
	{ T1 = [Tmul x T num1 x T num3 -> T0], Tf =[Tx X Ty -> T2] }
Tnum1 = number	
Tnum3 = number	
T mul = [Number X number -> Number]	
Tf = Tmul	
Tx = T num1	
Ty = T num 3	

Equation	Substitution
	{ T1 = [Tmul x number x T num3 -> T0], Tf =[Tx X Ty -> T2], T1 = [Tf x Tx x Ty - > T2] Tnum1 = number }
Tnum3 = number	
T mul = [Number X number -> Number]	
Tf = Tmul	
Tx = T num1	
Ty = T num 3	

Equation	Substitution
	{ T1 = [Tmul x number x number -> T0], Tf =[Tx X Ty -> T2], T1 = [Tf x Tx x Ty - > T2] Tnum1 = number, Tnum3 = number }
T mul = [Number X number -> Number]	
Tf = Tmul	
Tx = T num1	
Ty = T num 3	

Equation	Substitution
	{ T1 = [[Number X Number -> Number] x number x number -> T0], Tf =[Tx X Ty -> T2], T1 = [Tf x Tx x Ty - > T2] Tnum1 = number, Tnum3 = number Tmul = [Number X Number -> Number] }
Tf = Tmul	
Tx = T num1	
Ty = T num 3	

Equation	Substitution
	{ T1 = [[Number X Number -> Number] x number x number -> T0], Tf =[Tx X Ty -> T2], T1 = [Tf x Tx x Ty - > T2] Tnum1 = number, Tnum3 = number Tmul = [Number X Number -> Number], }
Tx = T num1	
Ty = T num 3	
Tx = number	
Ty = number	
T2 = number	

Equation	Substitution
	{ T1 = [[Number X Number -> Number] x number x number -> T0], Tf =[T num1 X Ty -> T2], T1 = [Tf x Tx x Ty - > T2] Tnum1 = number, Tnum3 = number Tmul = [Number X Number -> Number], Tx = T num1 }
Ty = T num 3	
Tx = number	
Ty = number	
T2 = number	

Equation	Substitution
	{ T1 = [[Number X Number -> Number] x number x number -> T0], Tf =[number X number -> T2], T1 = [Tf x Tx x Ty - > T2] Tnum1 = number, Tnum3 = number Tmul = [Number X Number -> Number], Tx = T num1, Ty = T num 3 }
T2 = number	

Equation	Substitution
	{ T1 = [[Number X Number -> Number] x number x number -> T0], Tf =[number X number -> number], T1 = [Tf x Tx x Ty - > T2] Tnum1 = number, Tnum3 = number Tmul = [Number X Number -> Number], Tx = T num1, Ty = T num 3, T2 = number }
T2 = number	

Equation	Substitution
	{ T1 = [[Number X Number -> Number] x number x number -> number], Tf =[number X number -> number], T1 = [Tf x Tx x Ty - > T2] Tnum1 = number, Tnum3 = number Tmul = [Number X Number -> Number], Tx = T num1, Ty = T num 3, T2 = T0 T2 = number }

((lambda (f1 x1 y1) (f1 x1 y1)) * 1 3) -> ((lambda (f x y) (f x y)) * 1 3) ->

((lambda ([f : (Number X number -> Number)] [x : Number] [Y: number]) : Number