1.(a)

let func i lst =

match lst with

|h::t -> if h = 1 && i = 1 then [true] else [false]

|[] -> [false]

1.(b)

let func a =

(fun a y -> y a) a

1.(c)

let f a b =

match a with

|(l,n) -> (b::l,n+b)

2.

let rec aux acc ts =

match ts with

|[] -> acc

|h::t ->

(match h with

|Leaf->aux acc t

|Node(l,x,r) -> aux (acc+x) (l::r::t));;

let rec sumtailrec t =

aux 0 [t];;

3.(a)true

3.(b)false

4.

(lambda x.x (lambda x.y x)) (lambda z.z)

Alpha convert to (lambda l.l (lambda x.y x)) lambda z.z

(lambda x.y x) (lambda z.z)

y (lambda z.z)

5.

(lambda x.lambda y.x y z) (lambda c.c) ((lambda a.a) b)

(lambda x.lambda y.x y z) (lambda c.c) (b)

(lambda y.(lambda c.c) y z) (b)

(lambda c.c) b z

b z

6.

(lambda x. (lambda y.(x y))) y

Alpha convert to (lambda x. (lamda z.(x z))) y

lamda z.(y z)

7.c

(Lambda x.(lambda y.y x)) a b

（lambda y.y a) b

b a