

```

case=A
input=
main = double 21;
double x = x + x;
g y z s = case a + 1 of <1>a b -> \g. a b s; <2>-> letrec x=fib; y=fact; z= x*y* (fib fact x y z) * z + y in x y;
f x = h;
h x = (let z = y in x z (a*3));
f a = a + 2 * 4
output=
[[[["main",[],EAp (EVar "double") (ENum 21)),("double",["x"],EAp (EAp (EVar "+") (EVar "x")) (EVar "x")),("g",["y","z","s"],ECase (EAp (EAp (EVar "+") (EVar "a")) (ENum 1)) [(1,["a","b"],ELam ["g"] (EAp (EAp (EVar "a") (EVar "b")) (EVar "s"))),(2,[],ELet Recursive [(("x",EVar "fib"),("y",EVar "fact"),("z",EAp (EAp (EVar "+") (EAp (EAp (EVar "*") (EVar "x")) (EAp (EAp (EVar "*") (EVar "y")) (EAp (EAp (EVar "*") (EAp (EAp (EAp (EAp (EVar "fib") (EVar "fact")) (EVar "x")) (EVar "y")) (EVar "z")))) (EVar "z")))) (EVar "y"))] (EAp (EVar "x") (EVar "y"))))),(("f",["x"],EVar "h"),("h",["x"],ELet NonRecursive [(("z",EVar "y")] (EAp (EAp (EVar "x") (EVar "z")) (EAp (EAp (EVar "*") (EVar "a")) (ENum 3))))),(("f",["a"],EAp (EAp (EVar "+") (EVar "a")) (EAp (EAp (EVar "*") (ENum 2)) (ENum 4))))],[""]]]
main = double 21;
double x = x + x;
g y z s = case a + 1 of
  <1>a b -> \g.(a b) s;
  <2>-> letrec
    x = fib;
    y = fact;
    z = (x * (y * (((fib fact) x) y) z) * z)) + y
  in x y;
f x = h;
h x = let
  z = y
  in (x z) (a * 3);
f a = a + (2 * 4)

```

```

case=C
input=
main = letrec x = x + x; y=z+1;
g111 = h*3*x;
f12 = case a+1 of <1> -> a+2;
<2> -> \a b . a+b
in z (a*3);
f a = h a * 5;
k a b = Pack{2,1} a;
k1 a b = let a = 2 in letrec b = 4 in a * b *5
output=
[[[("main",[],ELet Recursive [(("x",EAp (EAp (EVar "+") (EVar "x")) (EVar "x")),("y",EAp (EAp (EVar "+") (EVar
"z")) (ENum 1)),("g111",EAp (EAp (EVar "*" (EVar "h")) (EAp (EAp (EVar "*" (ENum 3)) (EVar
"x"))),("f12",ECase (EAp (EAp (EVar "+") (EVar "a")) (ENum 1)) [(1,[],EAp (EAp (EVar "+") (EVar "a")) (ENum
2)),(2,[],ELam ["a","b"] (EAp (EAp (EVar "+") (EVar "a")) (EVar "b")))]]) (EAp (EVar "z") (EAp (EAp (EVar "*"
(EVar "a")) (ENum 3))))),("f",["a"],EAp (EAp (EVar "*" (EAp (EVar "h") (EVar "a")))) (ENum
5)),("k",["a","b"],EAp (EConstr 2 1) (EVar "a"))],("k1",["a","b"],ELet NonRecursive [(("a",ENum 2)] (ELet
Recursive [(("b",ENum 4)] (EAp (EAp (EVar "*" (EVar "a")) (EAp (EAp (EVar "*" (EVar "b")) (ENum 5)))))),(""])]
main = letrec
  x = x + x;
  y = z + 1;
  g111 = h * (3 * x);
  f12 = case a + 1 of
    <1> -> a + 2;
    <2> -> \a b.a + b
  in z (a * 3);
f a = (h a) * 5;
k a b = Pack{2,1} a;
k1 a b = let
  a = 2
  in letrec
    b = 4
    in a * (b * 5)

```

```

case=D
input=
main = letrec x = x + x;
y=z+1;
g111 = h*3*x;
f12 = let a= h*3;
b=case a+1 of
<1> -> a+2;
<2> -> \a b . a+b
in z (a*3)
in a*b+c;
f a = h a * 5;
k a b = Pack{2,1} a;
k1 a b = let a = 2 in letrec b = 4 in a * b *5

```

output=

```

[[(["main",[],ELet Recursive [{"x",EAp (EAp (EVar "+") (EVar "x")) (EVar "x")),("y",EAp (EAp (EVar "+") (EVar "z")) (ENum 1)),("g111",EAp (EAp (EVar "*" (EVar "h")) (EAp (EAp (EVar "*" (ENum 3)) (EVar "x"))),("f12",ELet NonRecursive [{"a",EAp (EAp (EVar "*" (EVar "h")) (ENum 3)),("b",ECase (EAp (EAp (EVar "+") (EVar "a")) (ENum 1)) [(1,[],EAp (EAp (EVar "+") (EVar "a")) (ENum 2)),(2,[],ELam ["a","b"] (EAp (EAp (EVar "+") (EVar "a")) (EVar "b")))]]) (EAp (EVar "z") (EAp (EAp (EVar "*" (EVar "a")) (ENum 3))))] (EAp (EAp (EVar "+") (EAp (EAp (EVar "*" (EVar "a")) (EVar "b")) (EVar "c"))),("f",["a"],EAp (EAp (EVar "*" (EAp (EVar "h") (EVar "a")) (ENum 5)),("k",["a","b"],EAp (EConstr 2 1) (EVar "a"))),("k1",["a","b"],ELet NonRecursive [{"a",ENum 2}] (ELet Recursive [{"b",ENum 4}] (EAp (EAp (EVar "*" (EVar "a")) (EAp (EAp (EVar "*" (EVar "b")) (ENum 5))))),""))]]

```

```

main = letrec
  x = x + x;
  y = z + 1;
  g111 = h * (3 * x);
  f12 = let
    a = h * 3;
    b = case a + 1 of
      <1> -> a + 2;
      <2> -> \a b.a + b
    in z (a * 3)
  in (a * b) + c;
f a = (h a) * 5;
k a b = Pack{2,1} a;
k1 a b = let
  a = 2
  in letrec
    b = 4
    in a * (b * 5)

```

case=E

input=

f a b c = a \* b - c | a & b / 2; g a b = case a of <1> -> a & b & c; <2> c d -> a - b \* c / d

output=

```

[[(["f",["a","b","c"],EAp (EAp (EVar "|" (EAp (EAp (EVar "-") (EAp (EAp (EVar "*" (EVar "a")) (EVar "b")) (EVar "c")))) (EAp (EAp (EVar "&") (EVar "a")) (EAp (EAp (EVar "/" (EVar "b")) (ENum 2))))),("g",["a","b"],ECase (EVar "a") [(1,[],EAp (EAp (EVar "&") (EVar "a")) (EAp (EAp (EVar "&") (EVar "b")) (EVar "c"))), (2,["c","d"],EAp (EAp (EVar "-") (EVar "a")) (EAp (EAp (EVar "*" (EVar "b")) (EAp (EAp (EVar "/" (EVar "c")) (EVar "d")))))]),""))]]

```

f a b c = ((a \* b) - c) | (a & (b / 2));

g a b = case a of

<1> -> a & (b & c);

<2>c d -> a - (b \* (c / d))

case=F

input=

f1 x y z = x y y z;

f2 a b = case a b 2 of

<1> -> 1;

<2> -> 2

output=

```

[[(["f1",["x","y","z"],EAp (EAp (EAp (EVar "x") (EVar "y")) (EVar "y")) (EVar "z")),("f2",["a","b"],ECase (EAp (EAp (EVar "a") (EVar "b")) (ENum 2)) [(1,[],ENum 1),(2,[],ENum 2)]),""))]]

```

f1 x y z = (x y) y z;

f2 a b = case (a b) 2 of

```
<1> -> 1;  
<2> -> 2
```

case=G

input=

```
f x y = case x of <1> -> y ; <2> -> 3; g = \ a b . let c = a in c
```

output=

```
[([("f",["x","y"],ECase (EVar "x") [(1,[],EVar "y"),(2,[],ENum 3)]),("g",[],ELam ["a","b"] (ELet NonRecursive  
[("c",EVar "a")] (EVar "c")))),(""])
```

```
f x y = case x of
```

```
  <1> -> y;
```

```
  <2> -> 3;
```

```
g = \a b.let
```

```
  c = a
```

```
  in c
```

case=H

input=

```
f=3; g x y = let z=x in z; h x = case (let y=x in y) of <1> -> 2; <2> -> 5
```

output=

```
[([("f",[],ENum 3),("g",["x","y"],ELet NonRecursive [("z",EVar "x")] (EVar "z")),("h",["x"],ECase (ELet  
NonRecursive [("y",EVar "x")] (EVar "y")) [(1,[],ENum 2),(2,[],ENum 5)])),(""])
```

```
f = 3;
```

```
g x y = let
```

```
  z = x
```

```
  in z;
```

```
h x = case let
```

```
  y = x
```

```
  in y of
```

```
    <1> -> 2;
```

```
    <2> -> 5
```