REPORT

TEST CASES

a.SECD:

test cases

1.

```
let exp1 = Add (Num 1, Num 2);;
        answer list * table = ([N 3], [])
```

2.

```
let exp2 = Add (Num 1, Var "x");;
stkmc [] ["x",N 2] o [];;
- : answer list * table = ([N 3], [("x", N 2)])
```

3.

```
let exp3 = Lam ("x", Num 1);;
stkmc [] ["x",N 2] o [];;
- : answer list * table =
([Vcl ([("x", N 2)], "x", [NUM 1; RET])], [("x", N 2)])
```

4.

```
let exp4 = App (Lam ("x",Num 1), Num 1);;
stkmc [] ["x",N 2] o [];;
- : answer list * table = ([N 1], [("x", N 2)])
```

5.(Number of terms between 0 and 10)

```
let exp5 = App (Rlam ("x","f" ,( Ifthenelse (Iszero (Var "x"), Num 1, Add
(App(Var "f" , Add (Var "x", Num (-1))), Num 1)))),Num 10);;
stkmc [] ["x",N 2] o [];;
- : answer list * table = ([N 11], [("x", N 2)])
```

6. Find sum of numbers from 0 to n:

```
let exp6 = App (Rlam ("x","f" ,( Ifthenelse (Iszero (Var "x"), Num 0, Add
(App(Var "f" , Add (Var "x", Num (-1))), Var “x”)))),Var “y”);;
stkmc [] ["y",N 10] o [];;
- : answer list * table = ([N 56], [("y", N 10)])
```

7. Factorial

```
let exp7 = App (Rlam ("x","f" ,( Ifthenelse (Iszero (Var "x"), Num 1, Mul
(App(Var "f" , Add (Var "x", Num (-1))), Var “x”))),Var “y”);;
stkmc [] ["y",N 5] o [];;
- : answer list * table = ([N 120], [("y", N 5)])
```

8.Fib

```
let exp7 = App (Rlam ("x","f" ,( Ifthenelse (Iszero (Var "x"), Num 1,
Ifthenelse(Iszero(Add (Var “x”,Num (-1))),Num 1,(Add (App(Var "f" , Add
(Var "x", Num (-1))), App(Var "f" , Add (Var "x", Num (-2)))))))),Var “y”);;
stkmc [] ["y", N 10] o [];;
- : answer list * table = ([N 89], [("y", N 10)])
```

b.Krivine:

test cases

1.

```
let exp1 = Add (Num 1, Num 2);;
krivinemc [] [] o;;
- : closure = VL ([], N 3)
```

2.

```
let exp2 = Add (Num 1, Var "x");;
krivinemc [] ["x", (VL ([],N 3))] o;;
- : closure = VL ([("x", VL ([], N 3))], N 4)
```

3.

```
let exp3 = Lam ("x", Num 1);;
krivinemc [] ["x", (VL ([],N 3))] o;;
- : closure = FCL ([("x", VL ([], N 3))], "x", [NUM 1])
```

4.

```
let exp4 = App (Lam ("x",Num 1), Num 1);;
krivinemc [] [] o;;
- : closure = VL ([("x", CL ([], [NUM 1]))], N 1)
```

5.(Number of terms between 0 and 10)

```
let exp5 = App (Rlam ("x","f" ,( Ifthenelse (Iszero (Var "x"), Num 1, Add
(App(Var "f" , Add (Var "x", Num (-1))), Num 1)))),Num 10);;
krivinemc [] [] o;;
- : closure =
VL
 ([("x", CL ([], [NUM 10]));
   ("f",
    RFCL
     ([], "x", "f",
      [VAR "x"; ISZERO;
```

```
      COND ([NUM 1], [VAR "f"; APP [VAR "x"; NUM (-1); ADD]; NUM 1;
ADD])]))],
  N 11)
```

6. Find sum of numbers from 0 to n:

```
let exp6 = App (Rlam ("x","f" ,( Ifthenelse (Iszero (Var "x"), Num 0, Add
(App(Var "f" , Add (Var "x", Num (-1))), Var “x”))),Var “y”);;
krivinemc [] ["y", VL ([],N 10)] o;;
- : closure =
VL
 ([("x", CL ([("y", VL ([], N 10))], [VAR "y"]));
   ("f",
    RFCL
     ([("y", VL ([], N 10))], "x", "f",
      [VAR "x"; ISZERO;
       COND ([NUM 0], [VAR "f"; APP [VAR "x"; NUM (-1); ADD]; VAR "x";
ADD])]));
   ("y", VL ([], N 10))],
  N 55)
```

7. Factorial

```
let exp7 = App (Rlam ("x","f" ,( Ifthenelse (Iszero (Var "x"), Num 1, Mul
(App(Var "f" , Add (Var "x", Num (-1))), Var “x”))),Var “y”);;
krivinemc [] ["y", VL ([],N 5)] o;;
- : closure =
VL
 ([("x", CL ([("y", VL ([], N 5))], [VAR "y"]));
   ("f",
    RFCL
     ([("y", VL ([], N 5))], "x", "f",
      [VAR "x"; ISZERO;
       COND ([NUM 1], [VAR "f"; APP [VAR "x"; NUM (-1); ADD]; VAR "x";
MUL])]));
   ("y", VL ([], N 5))],
  N 120)
```

8.Fib

```
let exp7 = App (Rlam ("x","f" ,( Ifthenelse (Iszero (Var "x"), Num 1,
Ifthenelse(Iszero(Add (Var “x”,Num (-1))),Num 1,(Add (App(Var "f" , Add
(Var "x", Num (-1))), App(Var "f" , Add (Var "x", Num (-2)))))))),Var “y”);;
krivinemc [] ["y", VL ([],N 10)] o;;
- : closure =
VL
 ([("x", CL ([("y", VL ([], N 10))], [VAR "y"]));
   ("f",
    RFCL
     ([("y", VL ([], N 10))], "x", "f",
      [VAR "x"; ISZERO;
       COND
        ([NUM 1],
         [VAR "x"; NUM (-1); ADD; ISZERO;
          COND
           ([NUM 1],
            [VAR "f"; APP [VAR "x"; NUM (-1); ADD]; VAR "f";
             APP [VAR "x"; NUM (-2); ADD]; ADD])])]));
   ("y", VL ([], N 10))],
  N 89)
```