

A)

```

                                     Expression
-----
                                     if Expression then Expression else Expression
-----
                                     if Zero-exp (Expression) then Expression else Expression
-----
                                     if Zero-exp (Diff-exp (Expression, Expression)) then Expression else Expression
-----
                                     if Zero-exp (Diff-exp (Identifier, Expression)) then Expression else Expression
-----
                                     if Zero-exp (Diff-exp (Identifier, Const-exp)) then Expression else Expression
-----
if Zero-exp (Diff-exp (Identifier, Const-exp)) then Diff-exp (Expression, Expression) else Expression
-----
if Zero-exp (Diff-exp (Identifier, Const-exp)) then Diff-exp (Identifier, Expression) else Expression}
-----
if Zero-exp (Diff-exp (Identifier, Const-exp)) then Diff-exp (Identifier, Const-exp) else Expression
-----
    if Zero-exp (Diff-exp (Identifier, Const-exp)) then Diff-exp (Identifier, Const-exp) else Diff-exp
                                     (Expression, Expression)
-----
    if Zero-exp (diff-exp (Identifier, Const-exp)) then Diff-exp (Identifier, Const-exp) else Diff-exp
                                     (Identifier, Expression)
-----
    if Zero-exp (diff-exp (20, 12)) then Diff-exp (Identifier, Const-exp) else Diff-exp (Identifier,
                                     Expression)
-----
    if Zero-exp (8) then Diff-exp (Identifier, Const-exp) else Diff-exp (Identifier, Expression)
-----
    if #f then Diff-exp (Identifier, Const-exp) else Diff-exp (Identifier, Expression)
-----
    if #f then Diff-exp (10, 3) else Diff-exp (Identifier, Expression)
-----
    if #f then 7 else Diff-exp (Identifier, Expression)
-----
    if #f then 7 else Diff-exp (20, 15)
-----
    if #f then 7 else 5
-----
                                     5

```

B)

```

1} p1
2} [x = 12] p0
3} (expval->bool (zero? (value-of << -(x, 12) >> p1 ))
   (expval->bool (zero? ( -
                        (exp-val (value-of<<x>> p1))
                        (exp-val (value-of<<12>> p1))
                        )
   )
   #t
4} (value-of <<-( -(z, 3), -(y, z))>> p1)
5} -(z, 3))
6} -(y, z)

```

