Type Systems

Question 1. Consider the the following language definition based on WHILE:

```
(statements) s::= x := e | s1; s2
(expression) e::=   c | x | e1 + e2 | e1 * e2
(integer constant)  c
(integer variable)  x
(parity)   p::=  even | odd
```

Develop a type system for tracking the parity (even vs. odd) of integer expressions in this language. Assume that environment A is a map from each variable to even, odd, or top. Add the eight missing rules, each of which has judgments of the form `A |- e : p`, meaning that "under environment A, expression e has parity p." You may assume the following statements:

```
[c is an odd constant]    [c is an even constant]
----------------------    -----------------------     ----------------------
   A |- c : odd              A |- c : even                 A |- x : A(x)
```

In your answer, write each rule in the following form, where each [ ] is filled in with an appropriate statement.

```
  [       ]    [      ]
--------------------
       [      ]
```

a. Four rules for 'e1 + e2':

Answer:

```
A |- e1 : even        A |- e2 : even
-----------------------------------------
         A |- e1 + e2 : even


A |- e1 : even        A |- e2 : odd
-----------------------------------------
         A |- e1 + e2 : odd

A |- e1 : odd          A |- e2 : even
-----------------------------------------
         A |- e1 + e2 : odd

A |- e1 : odd       A |- e2 : odd
```

```
-------------------------------------
          A |- e1 + e2 : even
```

b. Four rules for ‘e1 * e2’:

Answer:

```
A |- e1 : even      A |- e2 : even
-----------------------------------------
           A |- e1 * e2 : even

A |- e1 : even       A |- e2 : odd
-----------------------------------------
           A |- e1 * e2 : even

A |- e1 : odd        A |- e2 : even
-----------------------------------------
          A |- e1 * e2 : even

A |- e1 : odd         A |- e2 : odd
-----------------------------------------
         A |- e1 * e2 : odd
```