```
Andrew Kaiser
Qla) Letter oft derivation
     <assign> -> <id>= <expr>/
             -> REvenue = Lexpr)
             -> Revenue = (term)
             -> Revenue = Leveny * (Fortor);
             -> Raine = <factor> * (factor);
             -> Perenue = (Lexpr) * LECCHON);
             -> Revenue = (Lexpr) + (tam) * Lfactor);
             -> Revenue = (Lterm) + Lterm>) * Lfortor);
             -> Perenul = (Lfactor) + Ltorny) * Lfactor);
            -> perenve = (Lie) + Lterm) *Lfactor);
            -> Revenue = (storeSale + (torm)) * (factor);
           -> Revenue = (store Soule + (term)* (factor)) * (factor);
           -) Percone = (storeSale + (factor) * Lforetor) * Lforetor);
           -> paenue = (storeSale + Lid) * Lfactor) * Lfactor);
           -> Revenue = (storeSale + onlineSale * Lfactor) * Lfactor)
           -> facine = CstorcSale + OhlicSale * (id>) * (factor);
           -> farme = (gtore Sale + online Sale * discount Rate > * Lifactor)
           -> Racine = (storeSale + onlieSale * discourt fete) * Lid);
```

-7 Raine = (store Sale + online Sale \* d. 3 count Rate) \* unit Price;

Q26) Parse True

