```
19-1-24
                                                                   25
   create a list of skolem constants
 infort re:
   dy fol to-cry(fol)
       statement = fol. suplace ("(=)", "_")
while '-' in statement:
            i = stalement. index ('-')
             new-statement = ['+ statement [: ]+'=)'+
                                     statement [i+1:] + ] & ['+
                                     statement[i+1:] + => '+
                                      statiment[:i] +']
     slatiment = statement. uplace ("=)", "-")

oxfor = \[([^]] +)\]

      statements = se. findall (onthe, statement)
      for i, 5 in courseale (statements):
           if I' in S. and 'J' not in S:
               statements[i]+=j
          statement = statement uplace(s, fol-to-CNF(s))
      for 5 in statements:
      while ' in statements:
          statement statement suplace (5;
                           -(['] 4' [' in statement
           i = stationent.index('-')
           nue statement - + statement (br: i) + 1 + statement [iti]
           stationent = stationent [: br] + new-stationent of br70 else
     while it in statiment:
          i = statement index ( it')
         statement [i], statement [i+2] = J', statement [i+2],
          statement = (1. join (s)
     statement = slatement . explace ( ~ [ \ Y', '[ N \ Y')
     stationent = stationent replace ('~[]','[~]')
      ([E| 4]) = npro
     statements = Re findall (expr., statement)
```

```
for s in statements:
              statiment = statement replace (s, fol-t.-enf(s))
           exps = `~\[[^]]+\]
          stationals = re findall (exps, state ment)
          for s in statements:
              statement = statement . explace (s, DeMorgan(s))
          ectus statement
         pent (skolomoption (fol-to-orf ("animal (y) (>) lovus (x, y)"))
                                       ("Vx [ + y [animal(y) => loves(x,y)
                                            => [ ]z[ (x,z) ]] [))
51.
                                       ( " [american (x) & weapon (y) &
                                          sells (x,y, z) & hostile (z)] =>
                                          cumnal(x)"))
    Output.
    [warinal(y) | Louis(x,y)] & [wloves(x,y)] animally)]
    [animal(q(x)) + whoses (x, q(x))] [ lonus (F(x), x)]
    [ warricon(x) | weapon(y) | wells (x,y,z) | whostile(z)]
                  whatther wiminal(x)
   Explanation
     to king(x) A guedy(x) => Emil(x)
          King (Richard) 1 Greedy (Richard) => Ervil (Richard).
  A 😂 B
        y Replace with → m (A ⇒ B) A (B ⇒ A)
                     ~ king (Richard) ungreedy (Richard) v Evil (Richard)
```

```
print(fol_to_cnf("bird(x)=>~fly(x)"))

print(fol_to_cnf("∃x[bird(x)=>~fly(x)]"))

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\neha2\OneDrive\Documents\NehaKamath_1BM21CS113_AILab> python ~bird(x)|~fly(x)
[~bird(A)|~fly(A)]
```