```
Convert POL sorto CENT program:
clef get Attusbuty (Stury):
       enpr = (([1)]+1))
       makhes = re. findall (expr, string)
       return [m to m in str (matches ) if m. 15/phill)
del get Predicales (String):
     exp = [[a-2~]+1([A-2a-2]+1)]
     return re. findall (cape string)
del De Morgan (sontence):
      String = " - join (list (sontace) - Copy(1)
      String = String. replace ("~~")
     Hag = "[ In string
     String = string replace (~~()(1)
     String = string . strip ("]")
     for predicate in get Predicals (string):
            String = String - replace (predicates + la (predicates)
    (prints) tail = 2
    tor is a in an omerak (shing):
        1) (== 11;
           ¿ ( j ) = ( b )
          els c== b1:
                 5[:]= 011
   String = (' . gosn (1)
    String = String . replace (" a. ~ 1)
```

```
rown & ([ { string }] il flog du string
def skolem zation (sentance):
   skolom-constants=[f'fchole]3' for a in vary
                                  (ovd ('A'), ovd (2)41)]
     Setukment = 11. joint (list (sent ence)-copy())
      makter = re. findall ("[+ ]].", Statmer )
      to make in marter [::-2]:
          Stukment = Stukment : riplua (murh)
          Starme = re. findall ('I[([1]]+\])) Skund)
 for s in Stukmeny.
              Statement = Statement. replace (3,5 [1:-1])
 ANTS. psp (01 3({ al [o] it len (al) elre march [17]))}
      vehin stukenet
 impour re
 de tol-to.cnt (tol):
          Statement = fol. replace (" = >11, (-1)
         cubile (_) in stutement;
              i = Statement . Indn ('-1)
             New - Stakment = "[' + Statement [: i] + (=) +
               Student [i + 1'] + '] & ['+ Stutement
                [;+1:]+'=) 1+ Skikment [: i]+'])
             Stutment = new - Stut emat
       Statement = Statement . replace ("=)", (-1)
       er-bx = (/[([,]]+)/],
        & tutement = re-tindall (corps , Stut comes)
```

```
for i, sin en-monate (statements):
             The " sas & " ) not ins!
                    Stukments [;]+= '] )
for sin statements:
              & twoment = Studement replace (s) tol to-
                                           cot (2)
new - Stukenet
       while '- +' in stubple!
           i = Stutement inch ('~+')
   Stubment = list (strokment)
     Studement (i) ) studement (i+1), Statement
                            [i+2] = (7) , Statement
                                           Ci+2],(0)
       while ~ 3 in statement.
            1 = +takement. inds: ("~3")
            S = list (statement)
            SCi), & [1+1], SCi+2] = [+1, SCi+2], A
            Student 2 ( ) . join (s)
       for Sin Stutements.
            Statement = Statement. replace (55 to Lto-ch)(s)
         CI+C[1]]/ ~ = cho
        Stukments = ve tipdall (empry stukment)
         tor sin stret emony:
              Shut ement = Statement ripplace (5,0 cM organ
       robyh Statement
```

```
Print (Skolomization (tol-to_cht (conimal(y) <=>

loves (2)411))

print (1kolomization (tol-to_cht (chi [ty Lunimal(y)

=) loves (2)4)])

print (tol-to-cht (famenium (1) to welpon (y) to

Sells (2)4,2) thositie(2)) =) cenimal(2)11))

Output

[manimal (4)1 loves (2)4) to male (2)7)

[ onimal (4)1 loves (2)4) to male (2)7)

[ conimal (4)1 loves (2)4) to male (2)7)

[ mamerican (2)1 mucopon (3) to sells (2)7,42) to mas (4)7)

Conimal (2)
```

## Output:

```
In [3]: print(Skolemization(fol_to_cnf("animal(y)<=>loves(x,y)")))
    print(Skolemization(fol_to_cnf("∀x[∀y[animal(y)=>loves(x,y)]]=>[∃z[loves(z,x)]]")))
    print(fol_to_cnf("[american(x)&weapon(y)&sells(x,y,z)&hostile(z)]=>criminal(x)"))

[~animal(y)|loves(x,y)]&[~loves(x,y)|animal(y)]
    [animal(G(x))&~loves(x,G(x))]|[loves(F(x),x)]
    [~american(x)|~weapon(y)|~sells(x,y,z)|~hostile(z)]|criminal(x)
```