Program - 9

Conversion of FOL to CNF

$$(P \Rightarrow Q) \wedge (S \Rightarrow P)$$

1. Eliminate Biconditionals with
2. " Implication P $\sim P \vee Q$
3. Apply De Morgan's laws and distribute negation inwards until it applies to atomic propositions.
4. Each variable must have unique name.
5. Replace $\exists$ existential quantifiers by Skolem functions.

Function

```
import re
def getAttributes (string):
    expr = '\([^)]+\)'
    matches = re.findall (expr, string)
    return [m for m in str (matches) if
        m.alpha()]


def getPredicates (string):
    expr = '[a-zv]+\([a-za-z,]+\)'
    return re.findall (expr, string)


def Skolemization (statement):
    skolem_constants = [f'{chr(c)}' for c
        in range (ord ('A'), ord ('Z')+1)]
```

```
    for match in matches [::-1]:
        statement = statement.replace(match,
    for predicate in getPredicates (statement):
        attribute = getAttributes (predicate)
    return statement


import re
def fol_to_cnf (fol):
    statement = fol.replace ('=>', '~')
    expr = '\([^)]+\)'

    statements = re.findall (expr, statement)
    for i, s in enumerate (statements):
        if '[' in s and ']' not in s:
            statements [i] += ']'

    for s in statements:
        statements += statement.replace (s,
                        fol_to_cnf (s))

    while '-' in statement:
        i = statement.index ()
        lhs = statement.index ('(') if '(' in
            statement else 0
    return Skolemization (statement)
```

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

Output:-

~bird (x) | ~fly (x)
[~ bird (A) | ~fly (A)]

```
print("Harshala Rani 1BM21CS074")
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)"))
```

```
Harshala Rani 1BM21CS074
[~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)
```