AI-LAB-TEST-2

1BM18CS027
Dhanraj .K.
29/12/20

Q) English sentence to FOL Manually and then int CNF.

Statement ⇒

' Every human beong has brain '

Code :-

```
import re
def fol_to_cnf(fol):
        Statement = fol.replace ("<=>", "_").
        while '_' in statement:
              i = statement .index('_')
              new_statement = '[' + Statement[:i] +
statement [i+1:]   + ']&[' + statement[i+1:]+'=>' +Statement[:i]+'
              Statement = new_statement.

        Statement = Statement . replace ("=>" , "-").
        expr ='\[([^]]+)\]'
        Statement = re. findall (expr, statement.
         for i, s in enumerate (statements):
              if '[' in s and. ']' not in s :
                    Statements[i] += ']'.

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

1 IBM18CS027
Dhanraj.K.

```
while '-' in statement:
        i = statement.index('-')
        br = statement.index('[') if '[' in stateme
else 0.
        new_statement = '~' + statement[br:i] + '|'
Statement .[i+1:]
        Statement = statement[:br] + new_statement
if br > 0 else new statement.
    while '~∀' in statement:
        i = statement.index('~∀')
        statement = list(statement)
        Statement[i], statement[i+1], statement
[i+2] = '∃', statement[i+2], '~'.
        Statement = ''.join(Statement)
    while '~∃' in statement:
        i = statement.index('~∃').
        S = list(statement).
        S[i], S[i+1], S[i+2] = '∀', S[i+2], '~'.
        Statement = ''.join(s).
    Statement = statement.replace('~[∀', '[~∀')
    Statement = statement.replace('~[∃', '[~∃')
    expr = '(~[∀|∃].)'
    statements = re.findall(expr, statement)
```

```
for s in Statements :
        Statement = Statement . replacl (s,DeMorgan(s))
    return Statement.
```

function to get attributes

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

to get predicates

```
  def  getPredicates (string ):
         expr  = '[a-z~]+\([A-Za-z,]+\)'
         return · re.findall (expr , String ).
```

de-morgan

```
  def  DeMorgan( sentence ):
          String  = ''.join (list (sentence).copy( )).
          String = String .replace ('~~','').
          flag : '[' in String.
          string = string . replace . ('~[','')
          String  = string . strip ('J').
```

1BM18CS024
Dhanraj K

```
for predicate in get Predicates (string);
        String = String . replace (Predicate, f'~{
Predicate}')
    s= list(String).
    for i, c in enumerate (string):
            if c=='|':
                    S[i]='&'.
            elif c=='&':
                    S[i]='|'.
    String = ''.join (s).
    String = String . replace ('~~' ;'').
    return f'[{String}]' if flag else String.

#take userinput as FOL and print CNF.

def main():
    for i in range(int(input("Enter number of FOL:")));
            print("- - - - - - - - - - - - -").
            fol = input ("Enter FOL statement:").
            print ("Conjunctive Normal Form:", end=
            print (Skolemization (fol_to_cnf( fol))).
            print("- - - - - - - - -")
main() .
```