Harshavardhana.k
IBM18CS035.
CSE 5A.

Artificial Intelligence

LAB-TEST-2

Program1.

Question: Unification program.

```
def getAttributes(expr):
    expr = expr.split("(")[1:]
    expr = "(".join(expr)
    expr = expr[:-1]
    expr = re.split("(?<!\(.),(?!.\))", expr)
    return expr

def replaceAttributes(expr, old, new):
    attr = getAttributes(expr)
    for index, val in enumerate(attr):
        if val == old:
            attr[index] = new
        predicate = getInitialPredicate(expr)
    return predicate + "(" + "," join(attr) + ")"

def unify(expr, exp2):
    if exp1 == exp2:
        return False.[]
    if isConstant(exp1) and isConstant(exp2):
        if exp1 != exp2
            return False
```

Harshavardhan.k
29/12/2020

AI-Lab-Test-2.

Harshavardhana.k.
1BM18CS035
CSE 5A.

```
    if isconstant (exp1):
        return [(exp2, exp1)]

    if isconstant (exp2):
        return [(exp2, exp1)]

      if isVariable (exp1):
          if checkOccars (exp1, exp2):
              return False
          else:
              return [(exp2, exp1)]

     if getInitialpredicate (exp1) ! = getInitial predicate (exp2)
             print ("predicate do not match. Cannot unifed")
             return False.

head1 = getFirstpart (exp1)
head2 = getFirst part (exp2)
initialsub = unify (head1, head2)
  if not initialsub:
        return False.
  if attribute Count ==1:
        return initialsub.

tail1 = getRemaining Part (exp1)
 tail2 = getRemainingpart (exp2)
  if initialsub != []:
       tail1 = apply (tail1, initialsub)
       tail2 = apply (tail2, initialsub)
 remaing sub = unify (tail1, tail2)
```

29/12/20
Harshavardhan.k

AI-Lab Test - 2.

Harshavardhana.k
1BM18CS035
CSE 5A.

```
initialsub.extend(remainingSub)
return initialsub.

def getRemainingPart(expr):
    predicate = getInitialpredicate( expr)
    atts = getAttributes(expr) -
    newExpr = predicate + "(" + "," .join (atts[1:]) +")"
    return new Expr

def getFirstpart(expr):
    atts = getAttributes(expr)
    return atts[0]

def checkOccurs(var, expr):
    if expr.find(var) == -1:
            return False
    return True.

def replaceAttribute(expr, old, new):
    atts = getAttributes(expr)
    for index, val in enumerate(atts):
        if vall == old:
            atts[index] = new
    predicate = getInitialpredicate(expr)
    return predicate + "(" + "," .join(atts) +")"
```

29/12/20
Harshavardhana.k