

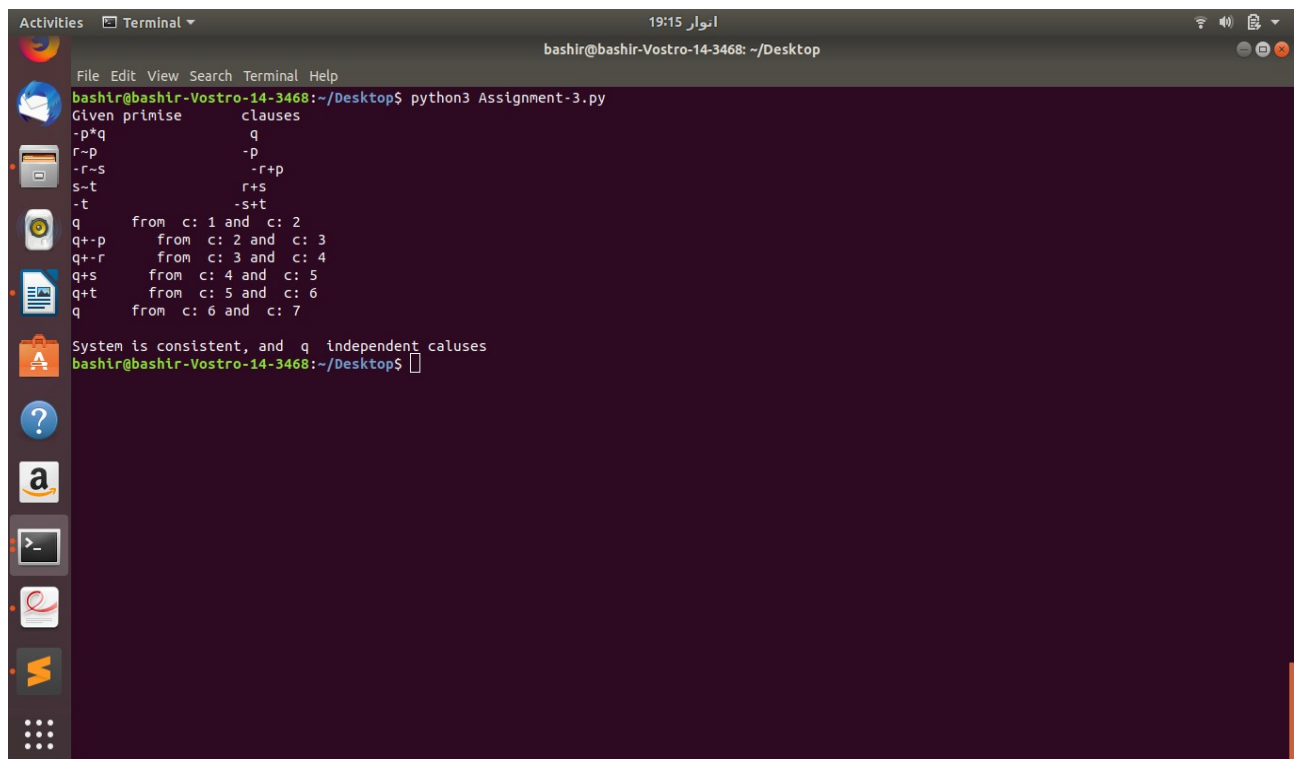
Assignment # 2

Hafiz Bashir Ahmad
p17-6079
section # B
DISCRETE STRUCTURE

PROGRAM INSTRUCTIONS:

1. implies sign “ \sim ”
2. conjunction sign “ $*$ ”
3. Disjunction sign “ $+$ ”
4. negation sign “ $-$ ”
4. clause from 1 and 2 by resolution save in clause 2 and so on

Example 6:



```
Activities Terminal 19:15 انوار
bashir@bashir-Vostro-14-3468: ~/Desktop
File Edit View Search Terminal Help
bashir@bashir-Vostro-14-3468:~/Desktop$ python3 Assignment-3.py
Given premise          clauses
-p*q                   q
r~p                    -p
-r~s                   -r+p
s~t                    r+s
-t                     -s+t
q                      from c: 1 and c: 2
q+-p                   from c: 2 and c: 3
q+-r                   from c: 3 and c: 4
q+s                    from c: 4 and c: 5
q+t                    from c: 5 and c: 6
q                      from c: 6 and c: 7

System is consistent, and q independent caluses
bashir@bashir-Vostro-14-3468:~/Desktop$
```

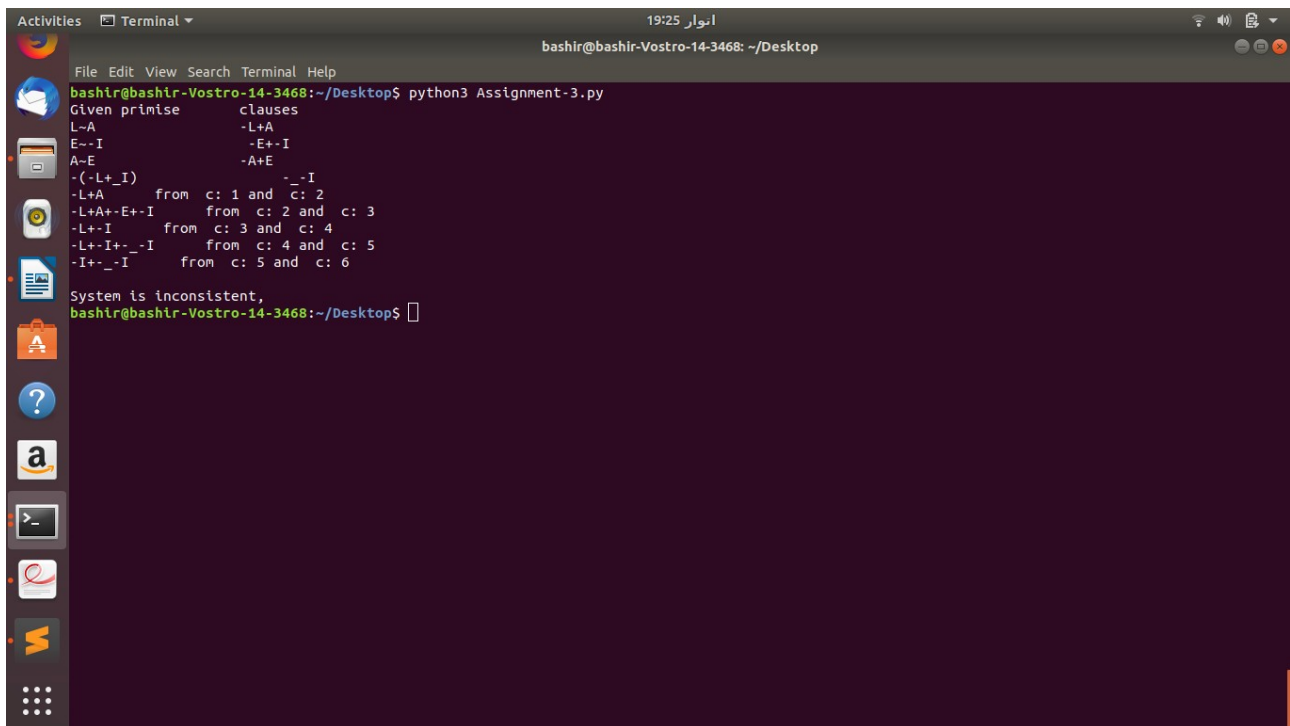
Example 7:

```
Activities Terminal 19:17 انوار
bashir@bashir-Vostro-14-3468: ~/Desktop
File Edit View Search Terminal Help
bashir@bashir-Vostro-14-3468:~/Desktop$ python3 Assignment-3.py
Given premise          clauses
-p~q                    p+q
-p~r                    p+r
-r~s                    r+s
-(q+s)                  -s
p+q      from c: 1 and c: 2
p+q+p+r  from c: 2 and c: 3
p+q+p+r+s from c: 3 and c: 4
p+q+p+r+r from c: 4 and c: 5
p+p+r+r  from c: 5 and c: 6
System is consistent
bashir@bashir-Vostro-14-3468:~/Desktop$
```

Example 8:

```
Activities Terminal 19:30 انوار
bashir@bashir-Vostro-14-3468: ~/Desktop
File Edit View Search Terminal Help
bashir@bashir-Vostro-14-3468:~/Desktop$ python3 Assignment-3.py
Given premise          clauses
T~M+V                  -T+M+V
S~-E                   -S+E
T*S                    S
-M                     T
-T+M+V      from c: 1 and c: 2
-T+M+V+-S+-E from c: 2 and c: 3
-T+M+V+-E      from c: 3 and c: 4
M+V+-E         from c: 4 and c: 5
V+-E           from c: 5 and c: 6
System is inconsistent,
bashir@bashir-Vostro-14-3468:~/Desktop$
```

Example 9:



```
bashir@bashir-Vostro-14-3468: ~/Desktop
File Edit View Search Terminal Help
bashir@bashir-Vostro-14-3468:~/Desktop$ python3 Assignment-3.py
Given premise          clauses
L~A                    -L+A
E~I                    -E+I
A~E                    -A+E
-(-L+I)                -_I
-L+A from c: 1 and c: 2
-L+A+-E+-I from c: 2 and c: 3
-L+-I from c: 3 and c: 4
-L+-I+-_I from c: 4 and c: 5
-I+-_I from c: 5 and c: 6
System is inconsistent,
bashir@bashir-Vostro-14-3468:~/Desktop$
```

The following program in Python 3:

```
#!/usr/bin/env python
# coding: utf-8
```

```
# In[422]:
```

```
def brackets_slove(s):
    if s[0] != "-":
        s = s.replace("(", "")
        s = s.replace(")", "")
        return s
    q = ""
    for idx, i in enumerate(s):
        if not(i == "-" or i == "(" or i == ")"):
            if i == "+":
                i = "*"
            q = q + i
        continue
```

```

    if i == "*":
        i = "+"
        q = q + i
        continue
    if s[idx-1] == "-":
        q = q + i
    else:
        q = q + "-" + i
return q

```

In[423]:

```

def implise(pr):
    li = pr.split('~')
    if "(" in li[0]:
        li[0] = brackets_slove(li[0])
    if "(" in li[1]:
        li[1] = brackets_slove(li[1])

    li[0] = "-" + "(" + li[0] + ")"
    li[0] = brackets_slove(li[0])

    pr = li[0] + "+" + li[1]
    return pr

```

In[424]:

```

def input_clauses():
    # string = "-p~q.-p~r.-r~s.-q~s" # Example 7
    # string = "-p*q.r~p.-r~s.s~t.t" # Example 6
    string = "T~M+V.S~-E.T*S.M"
    # string = "L~A.E~-I.A~E.L~_I"
    clauses = []
    s = string.split(".")

    if len(s[(len(s)-1)]) == 1 :
        s[(len(s)-1)] = "-" + s[(len(s)-1)]
    else:
        if "~" not in s[(len(s)-1)]:
            s[(len(s)-1)] = "-" + "(" + s[(len(s)-1)] + ")"
        else:
            s[(len(s)-1)] = implise( s[(len(s)-1)])
            s[(len(s)-1)] = "-" + "(" + s[(len(s)-1)] + ")"
    for idx , i in enumerate(s):
        if "~" in i:
            var = implise(i)
            clauses.append(var)
        else:
            if "(" in i:
                clauses.append(brackets_slove(i))
            else:
                clauses.append(i)
    clauses[len(clauses)-1] = clauses[len(clauses)-1].replace(")", "")
    for idx , i in enumerate(clauses):
        if "*" in i:

```

```

        i = i.split("*")
        clauses[idx] = i[0]
        clauses.insert(idx,i[1])
    print("Given premise      clauses")
    for i in range(0,len(s)):
        print(s[i] , "      ", clauses[i])
    return clauses

```

In[425]:

```
list1 = input_clauses()
```

In[426]:

```

def solve_clauses(f,s):
#   f = "-p+-r"
    f_list = []
#   s = "p+s"
    s_list = []
    final_clause = ""
    if f == s:
        return s
    if "+" in f:
        f_list = f.split("+")
    else:
        f_list.append(f)
    if "+" in s:
        s_list = s.split("+")
    else:
        s_list.append(s)
    for i in range(0,len(f_list)):
        if "-" in f_list[i]:
            var = f_list[i].replace("-", "")
            for j in range(0,len(s_list)):
                if var == s_list[j]:
                    s_list[j] = ""
                    f_list[i] = ""

        if "-" not in f_list[i]:
            var = "-" + f_list[i]
            for j in range(0,len(s_list)):
                if var == s_list[j]:
                    s_list[j] = ""
                    f_list[i] = ""
    for i in f_list:
        if i != "":
            final_clause = final_clause + "+" + i

    for i in s_list:
        if i != "":
            final_clause = final_clause + "+" + i
    final_clause = final_clause[1:]

    return final_clause

```

In[428]:

```
def apply_rule_of_resolution(list1):
    l = [""]
    count = 2;
    for i in range(0,len(list1)):
        var = solve_clauses(l[0], list1[i])
        print(var , " from c:", i+1 , "and c:" , count)
        count += 1
        l[0] = var
    if len(l) == 0 :
        print("\nSystem is consistent")
    if len(l) >= 2:
        print("\nSystem in not consisten")
    if len(l) == 1:
        print("\nSystem is inconsistent, and " , l[0] , " independent caluses")
```

In[429]:

```
apply_rule_of_resolution(list1)
```