Name: **SOURAV SAMANTA**
ASU ID: **1207860455** **Project 3 – User Manual**

##

Python Script: "**Project3.py**"

##

Python Version: 2.7

Author: Sourav Samanta
Python Packages Used: re
sys
numpy

##

Execution Steps:

The Program "**Project3.py**" operates in 2 modes:

```
E:\My Stuff\A S U\CEN 598\Programming Assignments\3\Samanta_Project3\Project3.py
-----------------------------------------------------------------
Option 1:       Batch Mode
Option 2:       Interactive Mode
-----------------------------------------------------------------
Enter Option:
```

Option 1: Batch Mode

The "**input.txt**" must be present in the current directory.

Local Disk (E:) ▸ My Stuff ▸ A S U ▸ CEN 598 ▸ Programming Assignments ▸ 3 ▸ Samanta_Project3

Name	Date modified	Type	Size
input.txt	11/14/2015 9:07 PM	TXT File	1 KB
Project3.py	11/23/2015 12:54 ...	PY File	29 KB

"input.txt"

input.txt

```
F = a.b.d + a.c.d + b.c + b.f + c.f
G = a.d.f + b.f + c.f + e.f
H = a.d.e + b.c + c.e
```

Console Output

```
-----------------------------------------------------------------------------
Option 1:     Batch Mode
Option 2:     Interactive Mode
-----------------------------------------------------------------------------
Enter Option:1
-----------------------------------------------------------------------------
Input SOP 1:        a.b.d + a.c.d + c.b + c.f + b.f
Unique Literals:    ['a', 'b', 'c', 'd', 'f']
Total Cubes:        [['a', 'b', 'd'], ['a', 'c', 'd'], ['b', 'c'], ['c', 'f'], ['b',
'f']]
.......................................
Row 1:        a.b.d
```

```
Row 2:       a.c.d
Row 3:       c.b
Row 4:       c.f
Row 5:       b.f
.......................................
Column 1:    a
Column 2:    b
Column 3:    c
Column 4:    d
Column 5:    f
.......................................

CUBE LITERAL MATRIX:
[[1 1 0 1 0]
 [1 0 1 1 0]
 [0 1 1 0 0]
 [0 0 1 0 1]
 [0 1 0 0 1]]
=-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-=
#(Kernel-Cokernel combinations): 5

-----------  [KERNELS] & [CO-KERNELS]  ------------
[a.c.d + c.f + a.b.d + c.b + b.f] & [1]
[c + b] & [a.d]
[a.d + f + c] & [b]
[a.d + b + f] & [c]
[b + c] & [f]
Global Loop Count: 35
****************************************
Input SOP 2:        e.f + a.d.f + b.f + c.f
Unique Literals:    ['a', 'b', 'c', 'd', 'e', 'f']
Total Cubes:        [['e', 'f'], ['a', 'd', 'f'], ['b', 'f'], ['c', 'f']]
.......................................
Row 1:       e.f
Row 2:       a.d.f
Row 3:       b.f
Row 4:       c.f
.......................................
Column 1:    a
Column 2:    b
Column 3:    c
Column 4:    d
Column 5:    e
Column 6:    f
.......................................

CUBE LITERAL MATRIX:
[[0 0 0 0 1 1]
 [1 0 0 1 0 1]
 [0 1 0 0 0 1]
 [0 0 1 0 0 1]]
=-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-=
#(Kernel-Cokernel combinations): 1

-----------  [KERNELS] & [CO-KERNELS]  ------------
[a.d + c + e + b] & [f]
Global Loop Count: 70
****************************************
Input SOP 3:        c.e + a.e.d + c.b
```

Name: **SOURAV SAMANTA**
ASU ID: **1207860455** **Project 3 – User Manual**

```
Unique Literals:    ['a', 'b', 'c', 'd', 'e']
Total Cubes:        [['c', 'e'], ['a', 'd', 'e'], ['b', 'c']]
.......................................
Row 1:       c.e
Row 2:       a.e.d
Row 3:       c.b
.......................................
Column 1:    a
Column 2:    b
Column 3:    c
Column 4:    d
Column 5:    e
.......................................

CUBE LITERAL MATRIX:
[[0 0 1 0 1]
 [1 0 0 1 1]
 [0 1 1 0 0]]
=-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-=
#(Kernel-Cokernel combinations): 3

-----------  [KERNELS] & [CO-KERNELS]  ------------
[a.e.d + c.e + c.b] & [1]
[b + e] & [c]
[c + a.d] & [e]
Global Loop Count: 100
****************************************

------------------ RESULTS ------------------

SOPs:
Input SOP 1: a.b.d + a.c.d + c.b + c.f + b.f
1:           a.b.d
2:           a.c.d
3:           c.b
4:           c.f
5:           b.f
----------------------------------------
Input SOP 2: e.f + a.d.f + b.f + c.f
8:           b.f
9:           c.f
6:           e.f
7:           a.d.f
----------------------------------------
Input SOP 3: c.e + a.e.d + c.b
10:          c.e
11:          a.e.d
12:          c.b
----------------------------------------
.......................................
SOP: 1 Row 1:a.d
SOP: 1 Row 2:b
SOP: 1 Row 3:c
SOP: 1 Row 4:f
SOP: 2 Row 5:f
SOP: 3 Row 6:c
SOP: 3 Row 7:e
.......................................
Column 1:    f
```

```
Column 2:    e
Column 3:    a.d
Column 4:    c
Column 5:    b
.......................................
CO-KERNEL CUBE MATRIX:
[[ 0  0  0  2  1]
 [ 5  0  1  3  0]
 [ 4  0  2  0  3]
 [ 0  0  0  4  5]
 [ 0  6  7  9  8]
 [ 0 10  0  0 12]
 [ 0  0 11 10  0]]

BINARY CO-KERNEL CUBE MATRIX:
[[0 0 0 1 1]
 [1 0 1 1 0]
 [1 0 1 0 1]
 [0 0 0 1 1]
 [0 1 1 1 1]
 [0 1 0 0 1]
 [0 0 1 1 0]]
=-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-=
```

Option 2: Interactive Mode

```
-------------------------------------------------------
Option 1:       Batch Mode
Option 2:       Interactive Mode
-------------------------------------------------------
Enter Option:   2
-------------------------------------------------------
-------------------------------------------------------
Sample SOP format:      c.b.d + a.b + ~b.c
Enter SOP 1:            a.b.d + a.c.d + b.c + b.f + c.f
=======================================================
Do you want to continue (Y/N)?  y
=======================================================
Enter SOP 2:            a.d.f + b.f + c.f + e.f
=======================================================
Do you want to continue (Y/N)?  n
=======================================================
```

Console Output

```
==============================================================================
Input SOP 1:        a.b.d + a.c.d + c.b + c.f + b.f
Unique Literals:    ['a', 'b', 'c', 'd', 'f']
Total Cubes:        [['a', 'b', 'd'], ['a', 'c', 'd'], ['b', 'c'], ['c', 'f'], ['b',
'f']]
.......................................
Row 1:       a.b.d
Row 2:       a.c.d
Row 3:       c.b
Row 4:       c.f
Row 5:       b.f
.......................................
Column 1:    a
Column 2:    b
Column 3:    c
Column 4:    d
Column 5:    f
.......................................
```

```
CUBE LITERAL MATRIX:
[[1 1 0 1 0]
 [1 0 1 1 0]
 [0 1 1 0 0]
 [0 0 1 0 1]
 [0 1 0 0 1]]
=-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-=
#(Kernel-Cokernel combinations): 5

-----------  [KERNELS] & [CO-KERNELS]  ------------
[a.c.d + c.f + a.b.d + c.b + b.f] & [1]
[c + b] & [a.d]
[a.d + f + c] & [b]
[a.d + b + f] & [c]
[b + c] & [f]
Global Loop Count: 35
****************************************
Input SOP 2:       e.f + a.d.f + b.f + c.f
Unique Literals:   ['a', 'b', 'c', 'd', 'e', 'f']
Total Cubes:       [['e', 'f'], ['a', 'd', 'f'], ['b', 'f'], ['c', 'f']]
.......................................
Row 1:       e.f
Row 2:       a.d.f
Row 3:       b.f
Row 4:       c.f
.......................................
Column 1:    a
Column 2:    b
Column 3:    c
Column 4:    d
Column 5:    e
Column 6:    f
.......................................

CUBE LITERAL MATRIX:
[[0 0 0 0 1 1]
 [1 0 0 1 0 1]
 [0 1 0 0 0 1]
 [0 0 1 0 0 1]]
=-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-=
#(Kernel-Cokernel combinations): 1

-----------  [KERNELS] & [CO-KERNELS]  ------------
[a.d + c + e + b] & [f]
Global Loop Count: 70
****************************************

------------------ RESULTS ------------------

SOPs:
Input SOP 1: a.b.d + a.c.d + c.b + c.f + b.f
1:           a.b.d
2:           a.c.d
3:           c.b
4:           c.f
5:           b.f
----------------------------------------
Input SOP 2: e.f + a.d.f + b.f + c.f
```

```
8:          b.f
9:          c.f
6:          e.f
7:          a.d.f
----------------------------------------
.......................................
SOP: 1 Row 1:a.d
SOP: 1 Row 2:b
SOP: 1 Row 3:c
SOP: 1 Row 4:f
SOP: 2 Row 5:f
.......................................
Column 1:    f
Column 2:    e
Column 3:    a.d
Column 4:    c
Column 5:    b
.......................................
CO-KERNEL CUBE MATRIX:
[[0 0 0 2 1]
 [5 0 1 3 0]
 [4 0 2 0 3]
 [0 0 0 4 5]
 [0 6 7 9 8]]

BINARY CO-KERNEL CUBE MATRIX:
[[0 0 0 1 1]
 [1 0 1 1 0]
 [1 0 1 0 1]
 [0 0 0 1 1]
 [0 1 1 1 1]]
=-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-=
```

KERNEL COMPUTATION LOGIC

The Kernel computation approach although recursive in nature has been modified suitably so that the same kernel doesn't gets computed again.

The high level logic is described below:

The **Row** and **Column** array combination corresponding to the computed **Kernel** is stored in a list. Each time the function to calculate the kernel is called, the input row and column array combination is checked for its uniqueness. If the values are not unique, it implies that the kernel has already been calculated for that combination of row and column. So, they need not be re-calculated and thus computation is reduced.

Code Snippet

```
446        # Step-3: Recursive call to make cube free
447        # With recursive check
448        if R2C2List not in self.kernel:
449            self.computeKernel(R2, C2)
450
```

The "`loopCount`" variable keeps track of the number of iterations.

Comparison of results of the number of iterations with and without recursive check:

There is a significant reduction in the number of "`Global Loop Count`" value when the recursive check is in place as compared to the recursive check being absent. (say: 31 vs 124 as shown in the below example.)

With recursive check: Lines 444 disabled but lines **448** and **449 enabled**

Name: **SOURAV SAMANTA**
ASU ID: **1207860455** **Project 3 – User Manual**

```
444 #             self.computeKernel(R2, C2)        # Without the check
445
446             # Step-3: Recursive call to make cube free and divide by other literals one at a time
447             # With recursive check
448             if R2C2List not in self.kernel:
449                 self.computeKernel(R2, C2)
450
```

```
--------------------------------------------------------------------------------
Option 1:    Batch Mode
Option 2:    Interactive Mode
--------------------------------------------------------------------------------
Enter Option:2
--------------------------------------------------------------------------------
Sample SOP format:  c.b.d + a.b + ~b.c
Enter SOP 1:        c.b.d + a.b + ~b.c
================================================================================
Do you want to continue (Y/N)?   n
================================================================================
Input SOP 1:        c.b.d + a.b + ~b.c
Unique Literals:    ['a', 'b', 'c', 'd', '~b']
Total Cubes:        [['b', 'c', 'd'], ['a', 'b'], ['c', '~b']]
........................................
Row 1:       c.b.d
Row 2:       a.b
Row 3:       c.~b
........................................
Column 1:    a
Column 2:    b
Column 3:    c
Column 4:    d
Column 5:    ~b
........................................

CUBE LITERAL MATRIX:
[[0 1 1 1 0]
 [1 1 0 0 0]
 [0 0 1 0 1]]
=-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-=
#(Kernel-Cokernel combinations): 3

-----------  [KERNELS] & [CO-KERNELS]  ------------
[c.~b + a.b + c.b.d] & [1]
[a + c.d] & [b]
[~b + b.d] & [c]
Global Loop Count: 31
****************************************
```

Without recursive check: Lines **444 enabled** but lines 448 and 449 disabled

```
444             self.computeKernel(R2, C2)        # Without the check
445
446             # Step-3: Recursive call to make cube free and divide by other literals one at a time
447             # With recursive check
448 #             if R2C2List not in self.kernel:
449 #                 self.computeKernel(R2, C2)
450
```

```
--------------------------------------------------------------------------------
Option 1:    Batch Mode
Option 2:    Interactive Mode
--------------------------------------------------------------------------------
Enter Option:2
```

Name: **SOURAV SAMANTA**
ASU ID: **1207860455** **Project 3 – User Manual**

```
-------------------------------------------------------------------------------
Sample SOP format:  c.b.d + a.b + ~b.c
Enter SOP 1:        c.b.d + a.b + ~b.c
===============================================================================
Do you want to continue (Y/N)?   n
===============================================================================
Input SOP 1:        c.b.d + a.b + ~b.c
Unique Literals:    ['a', 'b', 'c', 'd', '~b']
Total Cubes:        [['b', 'c', 'd'], ['a', 'b'], ['c', '~b']]
.......................................
Row 1:       c.b.d
Row 2:       a.b
Row 3:       c.~b
.......................................
Column 1:    a
Column 2:    b
Column 3:    c
Column 4:    d
Column 5:    ~b
.......................................

CUBE LITERAL MATRIX:
[[0 1 1 1 0]
 [1 1 0 0 0]
 [0 0 1 0 1]]
=-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-==-
#(Kernel-Cokernel combinations): 3

-----------  [KERNELS] & [CO-KERNELS]  ------------
[c.~b + a.b + c.b.d] & [1]
[a + c.d] & [b]
[~b + b.d] & [c]
Global Loop Count: 124
****************************************
```

TEST CASES

1. “input.txt” is missing

```
-----------------------------------------
Option 1:       Batch Mode
Option 2:       Interactive Mode
-----------------------------------------
Enter Option:   1
-----------------------------------------
=========================================
ERROR: 'input.txt' file was missing.
```

2. The SOPs in “input.txt” has invalid characters

input.txt

```
F = a.b.d + a.c.d + b.c + b.f + c.f.&.a
G = a.d.f + b.f + c.f + e.f
H = a.d.e + b.c + c.e
```

```
<terminated> E:\My Stuff\A S U\CEN 598\Programming
-----------------------------------------
Option 1:       Batch Mode
Option 2:       Interactive Mode
-----------------------------------------
Enter Option:   1
-----------------------------------------
Only a-z A-Z + . ~ 0 1 allowed.
```

3. The SOP is not represented in the correct format.

```
input.txt
1  F = a.b.d + ac.d + b.c + b.f + c.f.a
2  G = a.d.f + b.f + c.f + e.f
3  H = a.d.e + b.c + c.e
```

```
<terminated> E:\My Stuff\A S U\CEN 598\Programming
-----------------------------------------
Option 1:       Batch Mode
Option 2:       Interactive Mode
-----------------------------------------
Enter Option:   1
-----------------------------------------
'ac' is an Invalid Input!!!
```