1. Introduction:

A python code is written to find the prime cover of the function by using EXPAND operator of ESPRESSO algorithm. It has various definitions like to find list of variables, to write the function in PCN notation, to find the complement of the function using SHARP and INTERSECTION operators, to find the weights of the cubes, to find the blocking matrices of the cubes and to find the expanded cube.

2. Inputs:

There are two inputs that a user should input to plot the ROBDD.

1. On set of the Boolean function: It shall be given in SOP format with complemented variables as uppercase letters.
   E.g.: (ab + a'b') should be entered as (ab + AB)

2. DC set of the Boolean function: If there is no DC set for the function, then it is entered as n. Else it shall be given in SOP format with complemented variables as uppercase letters.

```
Enter the ON set: ab+bD+AbC+Bcd+AcD+ABcd
Enter the DC set. If there is no DC set, enter n. aBc
Cover of the function:  ['ab', 'bD', 'AbC', 'Bcd', 'AcD', 'ABcd', 'aBc']
```

3. Outputs:

First the function is converted into PCN notation for finding the complement.

```
Enter the ON set: ab+bD+AbC+Bcd+AcD+ABcd
Enter the DC set. If there is no DC set, enter n. aBc
Cover of the function:  ['ab', 'bD', 'AbC', 'Bcd', 'AcD', 'ABcd', 'aBc']
{'ab': [0, 1, 0, 1, 1, 1, 1, 1], 'bD': [1, 1, 0, 1, 1, 0, 1, 1], 'AbC': [1, 0, 0, 1, 1, 1, 1, 0], 'Bcd': [1, 1, 1, 0, 0, 1, 0, 1], 'A
cD': [1, 0, 1, 1, 1, 0, 0, 1], 'ABcd': [1, 0, 1, 0, 0, 1, 0, 1]}
```

Using this PCN notation, complement of the function is generated.

```
Enter the ON set: ab+bD+AbC+Bcd+AcD+ABcd
Enter the DC set. If there is no DC set, enter n. aBc
Cover of the function:  ['ab', 'bD', 'AbC', 'Bcd', 'AcD', 'ABcd', 'aBc']
{'ab': [0, 1, 0, 1, 1, 1, 1, 1], 'bD': [1, 1, 0, 1, 1, 0, 1, 1], 'AbC': [1, 0, 0, 1, 1, 1, 1, 0], 'Bcd': [1, 1, 1, 0, 0, 1, 0, 1], 'A
cD': [1, 0, 1, 1, 1, 0, 0, 1], 'ABcd': [1, 0, 1, 0, 0, 1, 0, 1]}
fnc_bar: ABDC+ABdC+ABC+Abdc+aBDC+aBC+aBdC+BDC+BdC+BC
```

Then the weights of cubes are found and they are sorted accordingly.

```
Enter the ON set: ab+bD+AbC+Bcd+AcD+ABcd
Enter the DC set. If there is no DC set, enter n. aBc
Cover of the function:  ['ab', 'bD', 'AbC', 'Bcd', 'AcD', 'ABcd', 'aBc']
{'ab': [0, 1, 0, 1, 1, 1, 1, 1], 'bD': [1, 1, 0, 1, 1, 0, 1, 1], 'AbC': [1, 0, 0, 1, 1, 1, 1, 0], 'Bcd': [1, 1, 1, 0, 0, 1, 0, 1], 'A
cD': [1, 0, 1, 1, 1, 0, 0, 1], 'ABcd': [1, 0, 1, 0, 0, 1, 0, 1]}
fnc_bar: ABDC+ABdC+ABC+Abdc+aBDC+aBC+aBdC+BDC+BdC+BC
Weights of the cubes:  [23, 24, 20, 20, 21, 17]
Sorted cover:  ['ABcd', 'AbC', 'Bcd', 'AcD', 'ab', 'bD']
```

For each cube of the sorted cover, blocking matrix is generated and minimum rows are selected such that each column has at least one '1' to find the expanded cube.

```
Enter the ON set: ab+bD+AbC+Bcd+AcD+ABcd
Enter the DC set. If there is no DC set, enter n. aBc
Cover of the function:  ['ab', 'bD', 'AbC', 'Bcd', 'AcD', 'ABcd', 'aBc']
{'ab': [0, 1, 0, 1, 1, 1, 1, 1], 'bD': [1, 1, 0, 1, 1, 0, 1, 1], 'AbC': [1, 0, 0, 1, 1, 1, 1, 0], 'Bcd': [1, 1, 1, 0, 0, 1, 0, 1], 'A
cD': [1, 0, 1, 1, 1, 0, 0, 1], 'ABcd': [1, 0, 1, 0, 0, 1, 0, 1]}
fnc_bar: ABDC+ABdC+ABC+Abdc+aBDC+aBC+aBdC+BDC+BdC+BC
Weights of the cubes:  [23, 24, 20, 20, 21, 17]
Sorted cover:  ['ABcd', 'AbC', 'Bcd', 'AcD', 'ab', 'bD']
Blockin matrix of cube  ABcd is:  [[0, 0, 0, 0, 1, 1, 1, 0, 0, 0], [0, 0, 0, 1, 0, 0, 0, 0, 0, 0], [1, 1, 1, 0, 1, 1, 1, 1, 1, 1], [1
, 0, 0, 0, 1, 0, 0, 1, 0, 0]]
Blockin matrix of cube  AbC is:  [[0, 0, 0, 0, 1, 1, 1, 0, 0, 0], [1, 1, 1, 0, 1, 1, 1, 1, 1, 1], [0, 0, 0, 1, 0, 0, 0, 0, 0, 0]]
Blockin matrix of cube  Bcd is:  [[0, 0, 0, 1, 0, 0, 0, 0, 0, 0], [1, 1, 1, 0, 1, 1, 1, 1, 1, 1], [1, 0, 0, 0, 1, 0, 0, 1, 0, 0]]
Blockin matrix of cube  AcD is:  [[0, 0, 0, 0, 1, 1, 1, 0, 0, 0], [1, 1, 1, 0, 1, 1, 1, 1, 1, 1], [0, 1, 0, 1, 0, 0, 1, 0, 1, 0]]
Blockin matrix of cube  ab is:  [[1, 1, 1, 1, 0, 0, 0, 0, 0, 0], [1, 1, 1, 0, 1, 1, 1, 1, 1, 1]]
Blockin matrix of cube  bD is:  [[1, 1, 1, 0, 1, 1, 1, 1, 1, 1], [0, 1, 0, 1, 0, 0, 1, 0, 1, 0]]
Prime cover:  ['Bc', 'bC', 'Bc', 'cD', 'ab', 'bD']
```

4. Other Outputs:

ON set: WYZ+wxYZ+xyz+wxyZ+WxZ

DC set: WXyZ+wXyZ

```
Enter the ON set: WYZ+wxYZ+xyz+wxyZ+WxZ
Enter the DC set. If there is no DC set, enter n. WXyZ+wXyZ
Cover of the function:  ['WYZ', 'wxYZ', 'xyz', 'wxyZ', 'WxZ', 'WXyZ', 'wXyZ']
{'WYZ': [1, 0, 1, 0, 1, 0, 1, 1], 'wxYZ': [0, 1, 1, 0, 1, 0, 0, 1], 'xyz': [1, 1, 0, 1, 0, 1, 0, 1], 'wxyZ': [0, 1, 0, 1, 1, 0, 0, 1], 'WxZ': [1, 0, 1, 1, 1
, 0, 0, 1]}
fnc_bar: wyzX+wYz+wYzx+wYzX+wzX+wYX+wYZX+WyzX+yzX+WYz+WYzx+WYzX+WzX+Yz+Yzx+YzX+zX
Weights of the cubes:  [16, 15, 15, 15, 18]
Sorted cover:  ['wxYZ', 'xyz', 'wxyZ', 'WYZ', 'WxZ']
Blocking matrix of cube  wxYZ is:  [[0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0], [1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1], [1, 0, 0, 0, 0
, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0], [1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1]]
Blocking matrix of cube  xyz is:  [[1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1], [0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0], [0, 0, 0, 0, 0,
 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]
Blocking matrix of cube  wxyZ is:  [[0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0], [1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1], [0, 1, 1, 1, 0
, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0], [1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1]]
Blocking matrix of cube  WYZ is:  [[1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0], [1, 1, 1, 1, 1,
 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]]
Blocking matrix of cube  WxZ is:  [[1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1], [1, 1, 1, 1, 1,
 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]]
Prime cover:  ['xZ', 'xy', 'xy', 'WZ', 'WZ']
```