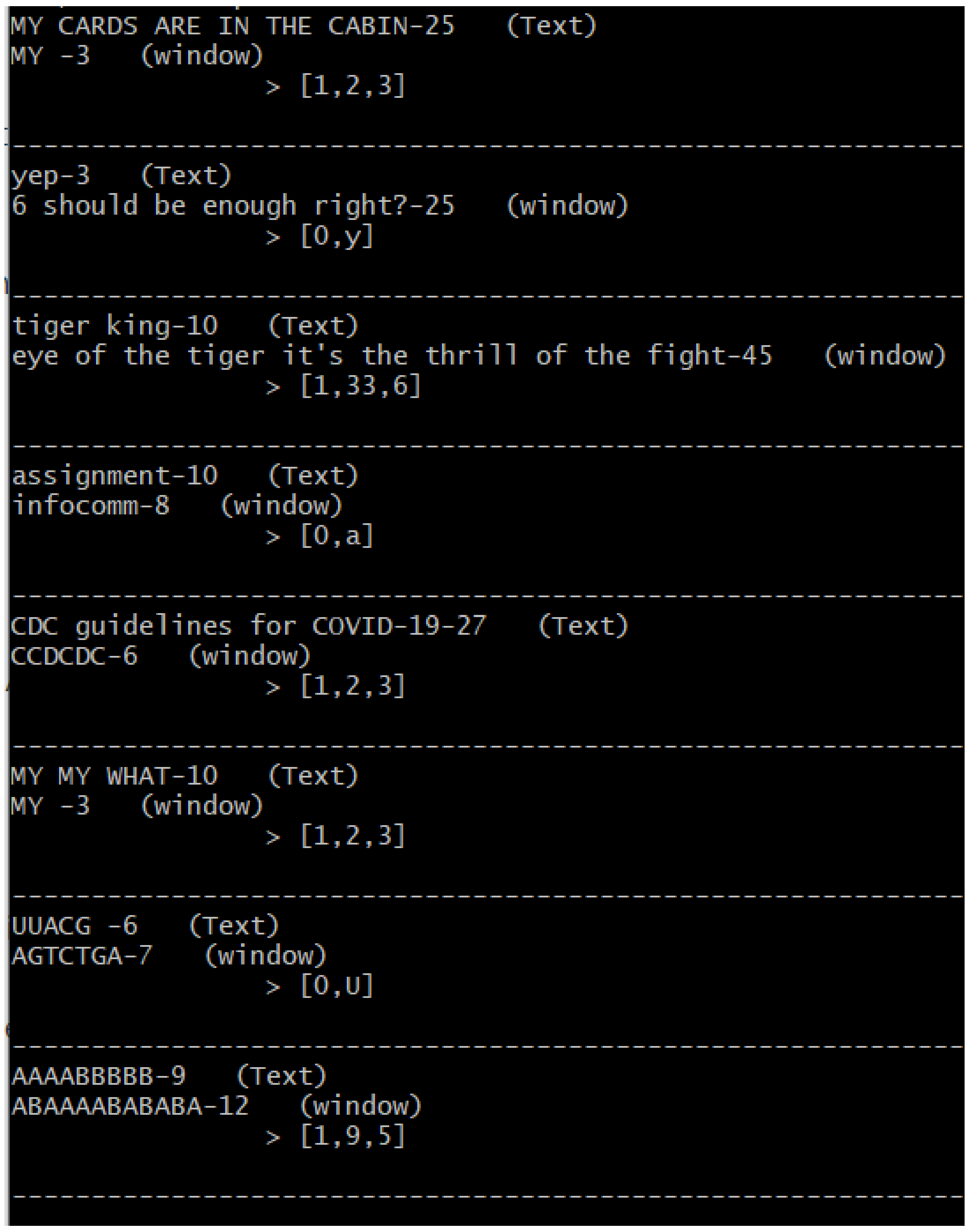
Assignment 1

- SRIHARI BANDARUPALLI
- 2021112006

Match Length Position

- Here we are taking window and text as inputs to this function.
- Then we are taking all possible substrings of the window and comparing it the Text
- Out of all the substrings that are matched we output the case where maximum length of substring is matched
- If none of the substrings match with the Input Text then we output the flag bit as 0 and the first character of the Text.
- Example

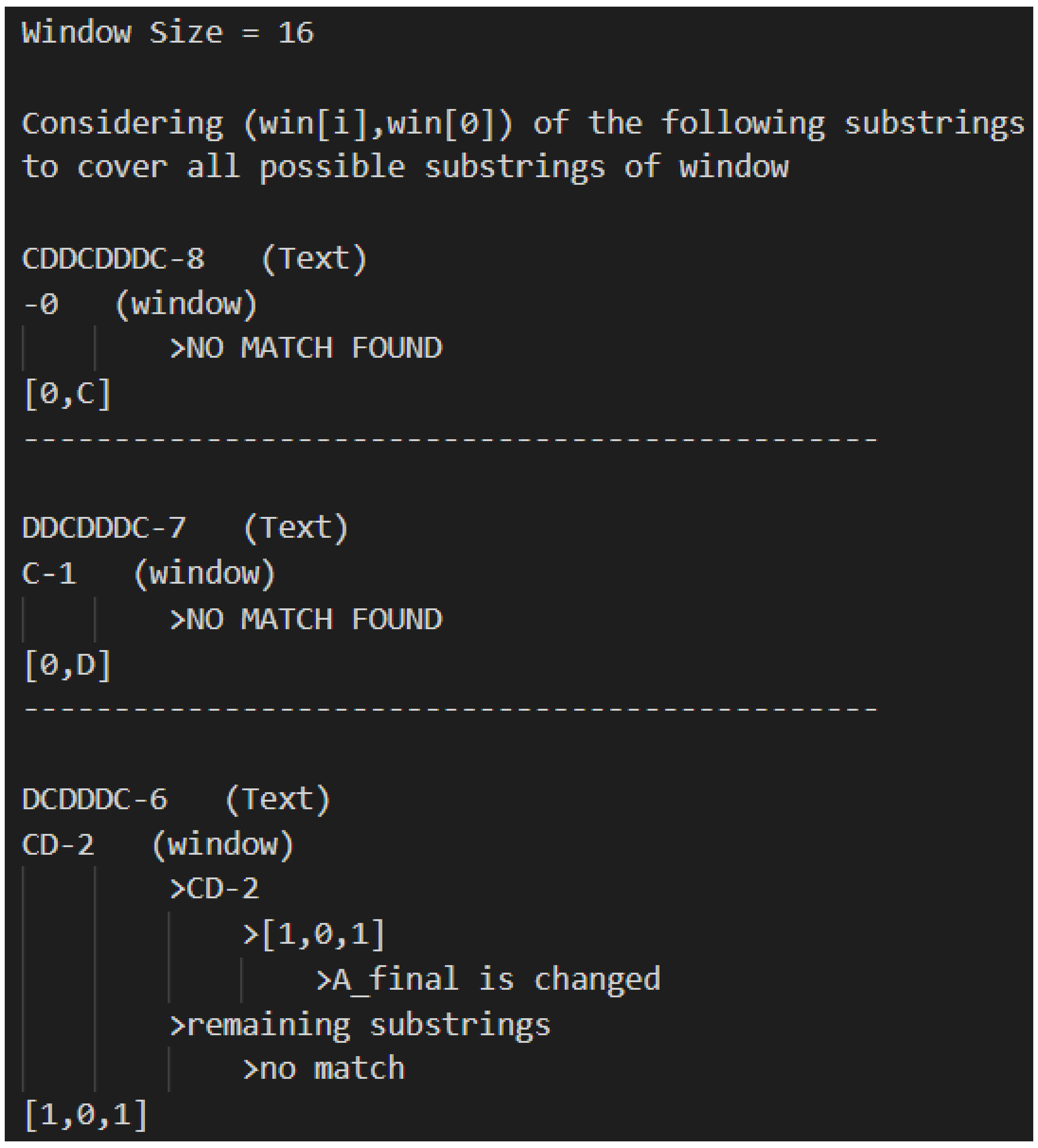
WINDOW	TEXT	OUTPUT
CCDCDC	CDC guidelines for COVID-19	[1,2,3]
ABAAAABABABA	AAAABBBBB	[1,9,5]
AGTCTGA	UUACG	[0,'U']



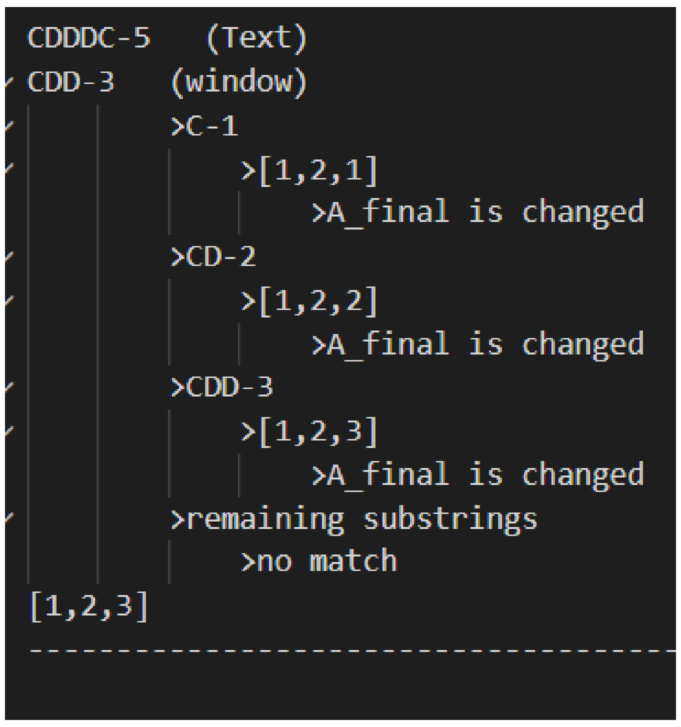
```
MY CARDS ARE IN THE CABIN-25   (Text)
MY -3   (window)
                > [1,2,3]

------------------------------------------------------------
yep-3   (Text)
6 should be enough right?-25   (window)
                > [0,y]

------------------------------------------------------------
tiger king-10   (Text)
eye of the tiger it's the thrill of the fight-45   (window)
                > [1,33,6]

------------------------------------------------------------
assignment-10   (Text)
infocomm-8   (window)
                > [0,a]

------------------------------------------------------------
CDC guidelines for COVID-19-27   (Text)
CCDCDC-6   (window)
                > [1,2,3]

------------------------------------------------------------
MY MY WHAT-10   (Text)
MY -3   (window)
                > [1,2,3]

------------------------------------------------------------
UUACG -6   (Text)
AGTCTGA-7   (window)
                > [0,U]

------------------------------------------------------------
AAAABBBBB-9   (Text)
ABAAAABABABA-12   (window)
                > [1,9,5]

------------------------------------------------------------
```

ParseSWLZ

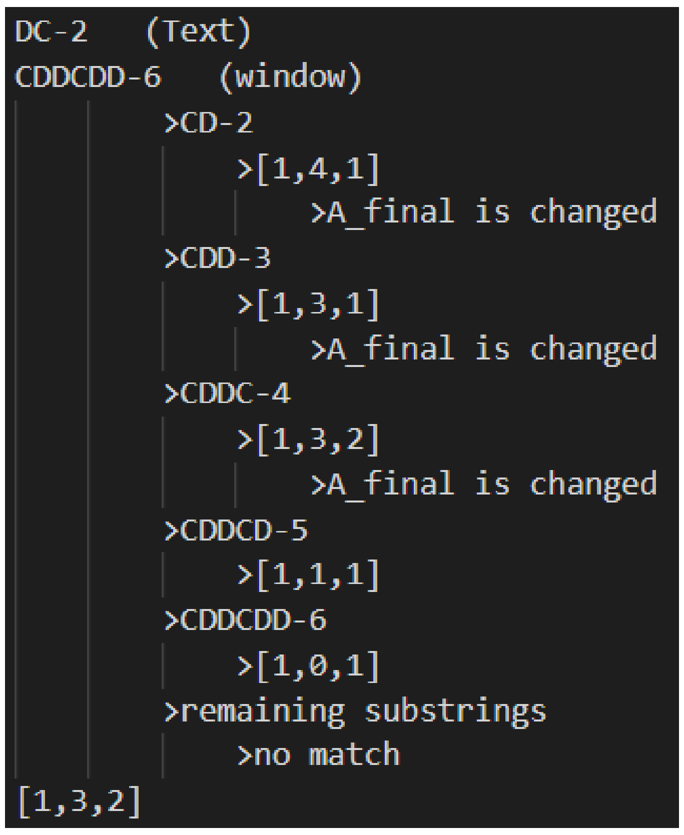
- Here we take the help of the previous function and Encode the Input Text
- We have to divide the Input Text into window and Text and then send it to the MatchLengthPosition to get the encoded part.
- We then shift the Encoded part into the Window and remaining part will the new Text which we send it to MatchLengthPostion function.
- Below is the working of my code



```
Window Size = 16

Considering (win[i],win[0]) of the following substrings
to cover all possible substrings of window

CDDCDDDC-8   (Text)
-0   (window)
        >NO MATCH FOUND
[0,C]
----------------------------------------------

DDCDDDC-7   (Text)
C-1   (window)
        >NO MATCH FOUND
[0,D]
----------------------------------------------

DCDDDC-6   (Text)
CD-2   (window)
        >CD-2
            >[1,0,1]
                >A_final is changed
        >remaining substrings
            >no match
[1,0,1]
```

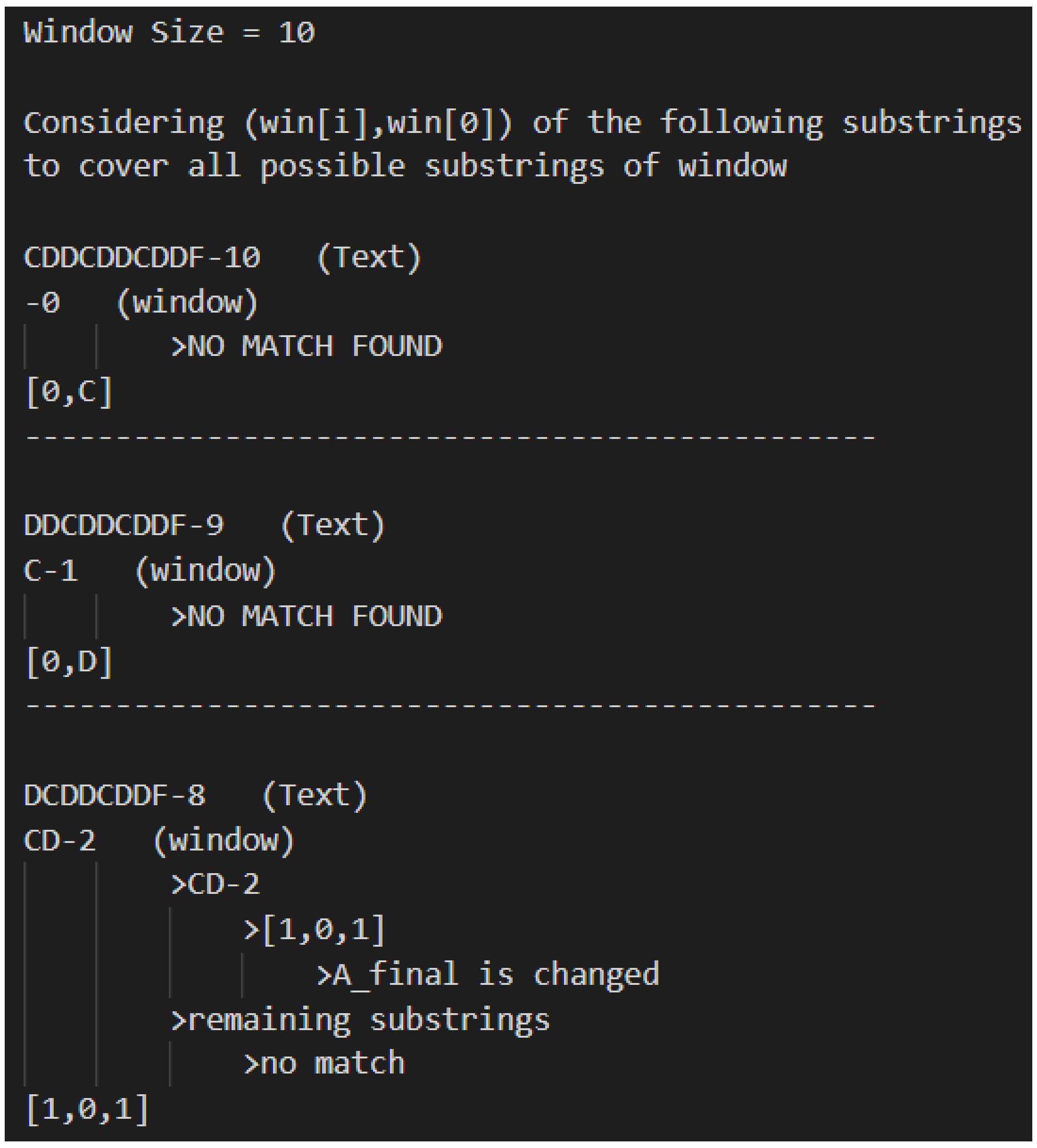


```
CDDDC-5   (Text)
CDD-3   (window)
        >C-1
            >[1,2,1]
                >A_final is changed
        >CD-2
            >[1,2,2]
                >A_final is changed
        >CDD-3
            >[1,2,3]
                >A_final is changed
        >remaining substrings
            >no match
[1,2,3]
-----------------------------------
```

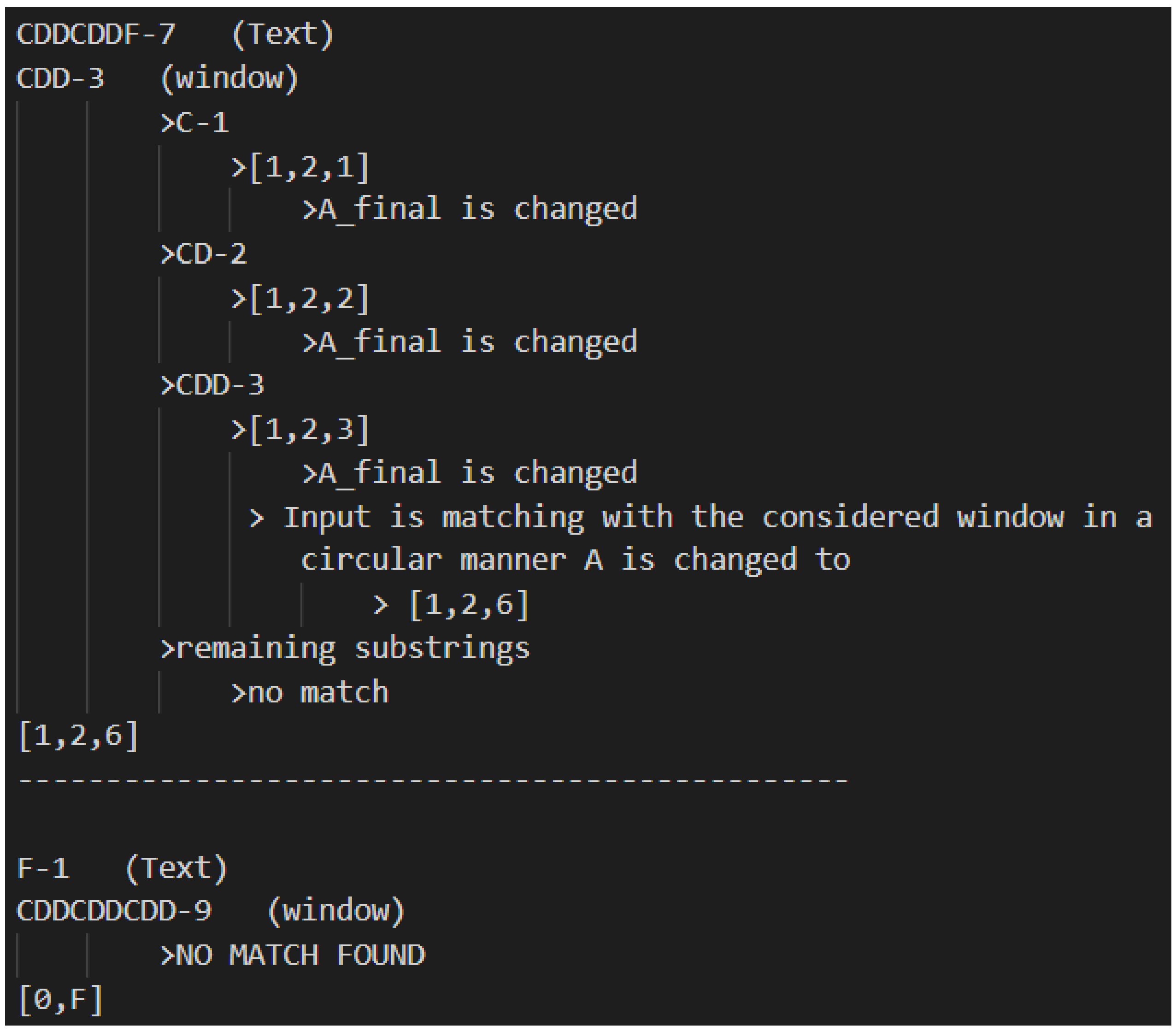


```
DC-2   (Text)
CDDCDD-6   (window)
        >CD-2
            >[1,4,1]
                >A_final is changed
        >CDD-3
            >[1,3,1]
                >A_final is changed
        >CDDC-4
            >[1,3,2]
                >A_final is changed
        >CDDCD-5
            >[1,1,1]
        >CDDCDD-6
            >[1,0,1]
        >remaining substrings
            >no match
[1,3,2]
```

- Cases where the no. of characters of our text is matching with the considered window if greater than the Window Length ,i.e. , characters of text are matching with characters of window in a circular manner
  - Example
    - INPUT TEXT = CDDCDDCDDDF
    - Window Size   = 10
    - OUTPUT = [0,C] [0,D] [1,0,1] [1,2,6] [0,F]



```
Window Size = 10

Considering (win[i],win[0]) of the following substrings
to cover all possible substrings of window

CDDCDDCDDF-10   (Text)
-0   (window)
        >NO MATCH FOUND
[0,C]
-----------------------------------------------

DDCDDCDDF-9   (Text)
C-1   (window)
        >NO MATCH FOUND
[0,D]
-----------------------------------------------

DCDDCDDF-8   (Text)
CD-2   (window)
        >CD-2
            >[1,0,1]
                >A_final is changed
        >remaining substrings
            >no match
[1,0,1]
```



```
CDDCDDF-7   (Text)
CDD-3   (window)
        >C-1
            >[1,2,1]
                >A_final is changed
        >CD-2
            >[1,2,2]
                >A_final is changed
        >CDD-3
            >[1,2,3]
                >A_final is changed
              > Input is matching with the considered window in a
                circular manner A is changed to
                    > [1,2,6]
        >remaining substrings
            >no match
[1,2,6]
-----------------------------------------------

F-1   (Text)
CDDCDDCDD-9   (window)
        >NO MATCH FOUND
[0,F]
```