Assignment 2

1) FOR FIRST INTERFACE:

Printing stats uptil now:

```
 Modified length of the text  :  222549


*Infected Weblinks are:*

     ftp://foo.bar/bla
     http://223.255.255.254
     http://a.b-c.de
     http://142.42.1.1:8080/
     http://foo.com/blah_(wikipedia)#cite-1
     https://foo.com/blah_blah
     http://foo.bar/?q=Test%20URL-encoded%20stuff
     http://www.example.com/wpstyle/?p=364
     https://www.example.com/foo/?bar=baz&inga=42&quux
     http://foo.com/blah_blah_(wikipedia)
     http://foo.com/blah_blah/
     http://1337.net
     http://142.42.1.1/
     http://foo.com/blah_blah_(wikipedia)_(again)
     http://foo.com/(something)?after=parens
     http://code.google.com/events/#&product=browser
     http://foo.com/blah_(wikipedia)_blah#cite-1
     ftp://foo.bar/baz
     http://foo.com/blah_blah
     http://j.mp


 Time taken to remove Infected URL's and multiple blank spaces  :  275.885582 milliseconds
```

Original length of the text : 224965

2) FOR SECOND INTERFACE:

By varying Pattern size:

Text range is fixed : 0 - 10000

ALGO USED / Pattern Length	KMP (Exec time : ms)	Rabin-karp (Exec time : ms)	Suffix Arrays (Exec time : ms)	
3	4.694461	18.284559	241.348743	
4	5.228043	17.512467	248.479366	
5	5.019904	18.108129	247.903108	
7	5.198479	17.207384	211.688996	
8	3.96069	19.058466	251.670122	

By varying Text length:

Pattern size is fixed : 4 (Wolf)

ALGO USED / Text Length	KMP (Exec time : ms)	Rabin-karp (Exec time : ms)	Suffix arrays (Exec time : ms)
10	0.034809	0.045061	0.305891
100	0.079393	0.226975	2.724886
1000	0.2985	1.490116	29.981136
10000	5.208966	17.596245	249.911881
100000	34.789563	110.603809	4080.501318

3) FOR THIRD INTERFACE:

OUTPUT:

Printing stats uptil now:

Build cross index using KMP : 10544.245005 milliseconds

Build cross index using Rabin-karp : 25865.430832 milliseconds

Build cross index using suffix array : 917907.422066 milliseconds

----
----

4) FOR FOURTH INTERFACE:

OUTPUT

```
-------
**********
Enter the max length:6
Enter the starting index:0
Enter the ending index:96000
gnizing  ->   Index range  ->  ( 2238 , 2245 )
Time to find Maximal palindromes  :  34.080982 milliseconds

**********
Enter the max length:7
Enter the starting index:0
Enter the ending index:200000
gnizing  ->   Index range  ->  ( 2238 , 2245 )
low-wol  ->   Index range  ->  ( 110081 , 110088 )
gnizing  ->   Index range  ->  ( 113420 , 113427 )
ecipice  ->   Index range  ->  ( 127021 , 127028 )
terpret  ->   Index range  ->  ( 136840 , 136847 )
ecipice  ->   Index range  ->  ( 178995 , 179002 )
ecipice  ->   Index range  ->  ( 188902 , 188909 )
terpret  ->   Index range  ->  ( 199196 , 199203 )
Time to find Maximal palindromes  :  49.465895 milliseconds
```


<u>5) FOR FIFTH INTERFACE:</u>

OUTPUT

<u>Printing stats uptil now:</u>

```
     Rabin-karp execution time  :  92.734098 milliseconds

     Time to find Maximal palindromes  :  73.927402 milliseconds

     Suffix Array Pattern searching time  :  0.054598 milliseconds

     Modified length of the text  :  222549

     Time taken to remove Infected URL's and multiple blank spaces  :  255.821705 milliseconds

     Pre Processing Suffix Array time  :  4214.547396 milliseconds

    Infected Weblinks are:
         ftp://foo.bar/bla
         http://223.255.255.254
         http://a.b-c.de
         http://142.42.1.1:8080/
         http://foo.com/blah_(wikipedia)#cite-1
         https://foo.com/blah_blah
```

```
        http://foo.bar/?q=Test%20URL-encoded%20stuff
        http://www.example.com/wpstyle/?p=364
        https://www.example.com/foo/?bar=baz&inga=42&quux
        http://foo.com/blah_blah_(wikipedia)
        http://foo.com/blah_blah/
        http://1337.net
        http://142.42.1.1/
        http://foo.com/blah_blah_(wikipedia)_(again)
        http://foo.com/(something)?after=parens
        http://code.google.com/events/#&product=browser
        http://foo.com/blah_(wikipedia)_blah#cite-1
        ftp://foo.bar/baz
        http://foo.com/blah_blah
        http://j.mp

    KMP Pattern searching time  :  40.084839 milliseconds

    Original length of the text  :  224965

    Pre Processing time for Rabin-karp  :  0.061989 milliseconds

    Pre processing time for KMP  :  0.021458 milliseconds
```

Learning Outcomes :

1) We came to know that Among KMP , Suffix arrays and Rabin karp , KMP is the most efficient among them.

2) The time required by the execution of suffix array pattern matching mainly goes to its preprocessing .

3) We understood that the module of re in python uses (FSA) i.e they use FSA based algorithms

4) We came to realization that pattern searching time remains almost constant for suffix arrays as its complexity is only O(log n)

5) Ultimately we thoroughly Enjoyed Our Assignment and got to know much more features in Python language.

Thanking you ,

Sriharsha Hatwar (01FB14ECS250)

Vishwambhara (01FB14ECS291)