

Assignment 8

ELP - 718 Telecom Software Laboratory

Kartik Gupta

2018JTM2250

2018-2020

A report presented for the assignment on
Python / Github



Bharti School Of
Telecommunication Technology and Management
IIT Delhi
India
September 27, 2018

Contents

1	Problem Statement-1	4
1.1	Problem Statement	4
1.2	Input Format	5
1.3	Output Format	5
1.4	Program Structure	6
1.5	Algorithm and implementation	7
1.6	TestCases	7
1.7	Screenshots	7
2	Problem Statement-2	8
2.1	Problem Statement	8
2.2	Input Format	8
2.3	Output Format	8
2.4	Program Structure	9
2.5	Algorithm and implementation	10
2.6	TestCases	10
2.7	Screenshots	11
3	Appendix	12
3.1	Code for ps1.py	12
3.2	Code for ps2.py	13

List of Figures

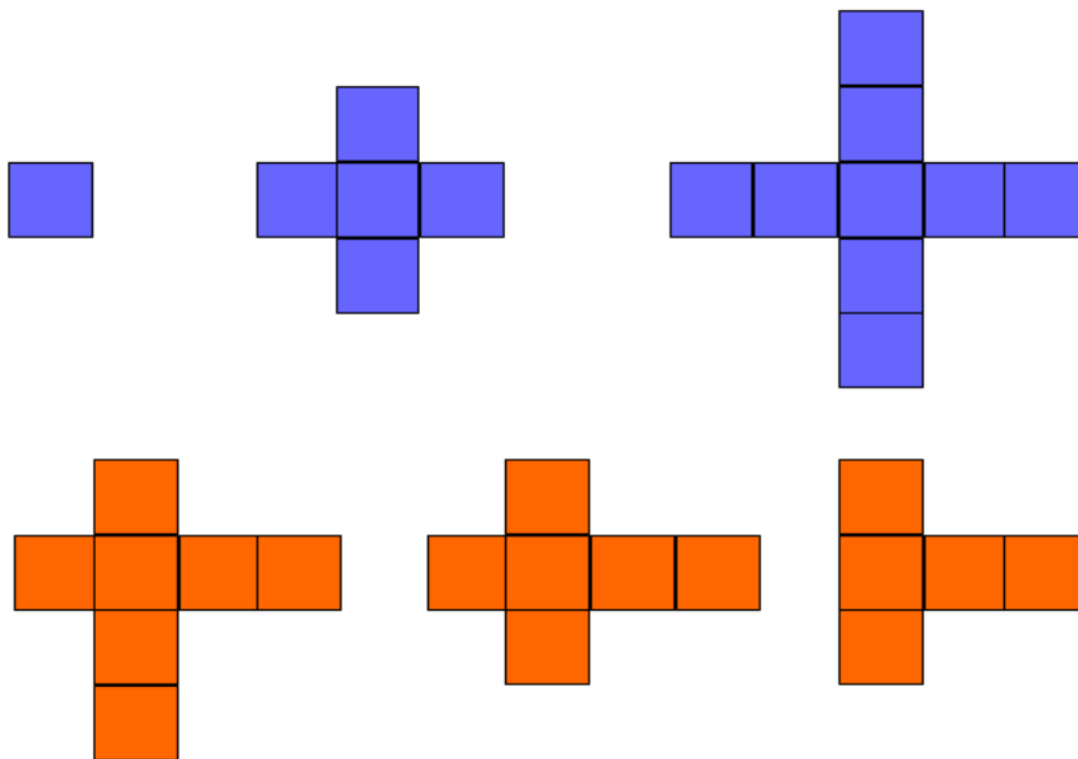
1	Flowchart for Program 1	6
2	Problem 1 Result	7
3	Flowchart for Program 2	9
4	Problem 2 Result	11

1 Problem Statement-1

1.1 Problem Statement

IIT Delhi, has just got the strongest computer. The professors in charge wants to check the computational capacity of the computer. So, they decided to create the problem which is to be given as an assignment to students. Can you help the professor to check the computation capability of the computer?

A valid cross is defined here as the two regions (horizontal and vertical) of equal lengths crossing over each other. These lengths must be odd, and the middle cell of its horizontal-region must cross the middle cell of its vertical-region.



In the diagram above, the blue crosses are valid and the orange ones are not valid.

Find the two largest valid crosses that can be drawn on smart cells in the grid, and return two integers denoting the dimension of the each of the two largest valid crosses. In the above diagrams, our largest crosses have dimension of 1, 5 and 9 respectively .

1.2 Input Format

The first line contains two space-separated integers, n and m .

Each of the next n lines contains a string of m characters where each character is either S(Smart) or D (Dull). These strings represent the rows of the grid. If the j th character in the i th line is S, then (i,j) is a cell smart. Otherwise it's a dull cell.

1.3 Output Format

Find two valid crosses that can be drawn on smart cell of the grid, and return the dimension of both the crosses in the reverse sorted order(i.e. First Dimension should be the larger one and other should be smaller one).

1.4 Program Structure

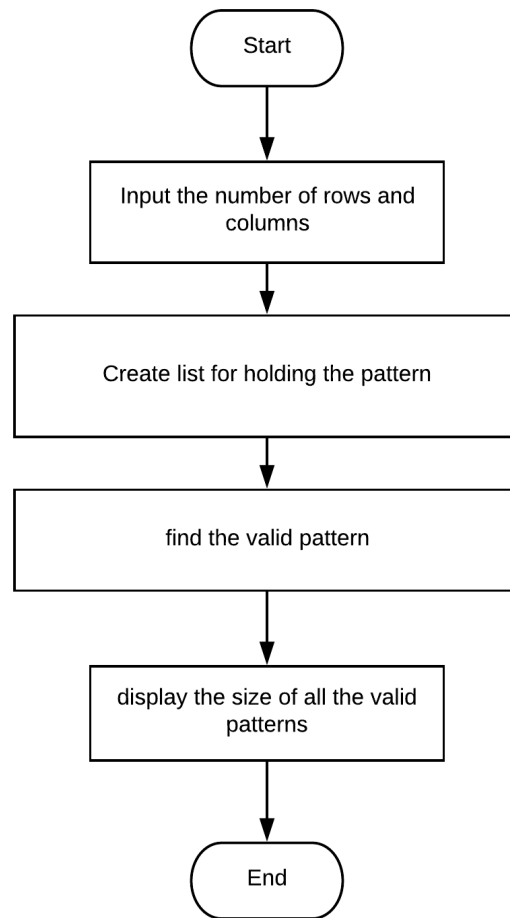


Figure 1: Flowchart for Program 1

1.5 Algorithm and implementation

- Input the number of rows and columns
- Create list for holding the pattern
- find the valid pattern
- display the size of all the valid patterns

1.6 TestCases

Sample Input 0

5 6

SSSSSS

SDDSDS

SSSSSS

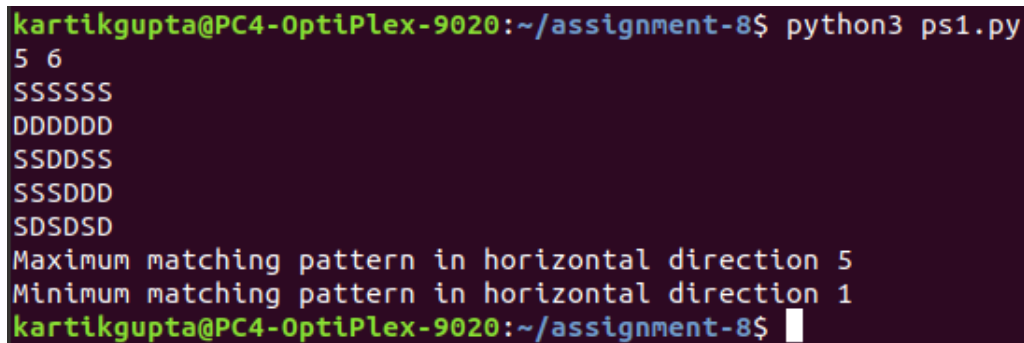
SSDDSD

SSSSSS

Sample Output 0

5 1

1.7 Screenshots



```
kartikgupta@PC4-OptiPlex-9020:~/assignment-8$ python3 ps1.py
5 6
SSSSSS
DDDDDD
SSDDSS
SSSDDD
SDSDSD
Maximum matching pattern in horizontal direction 5
Minimum matching pattern in horizontal direction 1
kartikgupta@PC4-OptiPlex-9020:~/assignment-8$
```

Figure 2: Problem 1 Result

2 Problem Statement-2

2.1 Problem Statement

After, getting mix results of valid crosses, professors decided to test the computation abilities on one more problem. This time professors wanted to test the decryption capabilities of the computer.

Encryption of a message requires three keys, k_1 , k_2 , and k_3 . The 26 letters of English and underscore are divided in three groups, [a-i] form one group, [j-r] a second group, and everything else ([s-z] and underscore) the third group. Within each group the letters are rotated left by k_i positions in the message. Each group is rotated independently of the other two. Decrypting the message means doing a right rotation by k_i positions within each group.

2.2 Input Format

All input strings comprises of only lowercase English alphabets and underscores.

2.3 Output Format

For each encrypted message, the output is a single line containing the decrypted string.

2.4 Program Structure

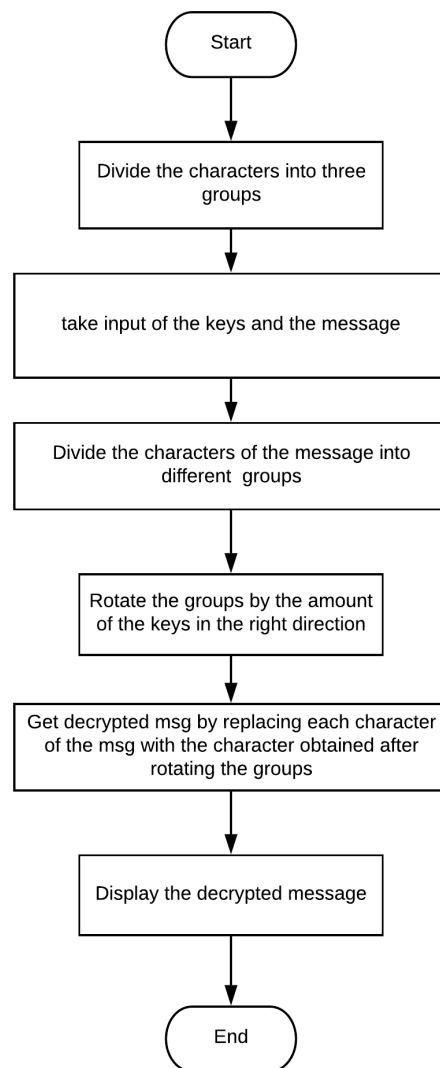


Figure 3: Flowchart for Program 2

2.5 Algorithm and implementation

- Divide the characters into three groups
- take input of the keys and the message
- Divide the characters of the message into different groups
- Rotate the groups by the amount of the keys in the right direction
- Get decrypted msg by replacing each character of the msg with the character obtained after rotating the groups
- Display the decrypted message

2.6 TestCases

Sample Input 0

2 3 4

dikhtkor_ey_tec_ocsusrsw_ehas_

Sample Output 0

hardwork_is_the_key_to_success

Sample Input

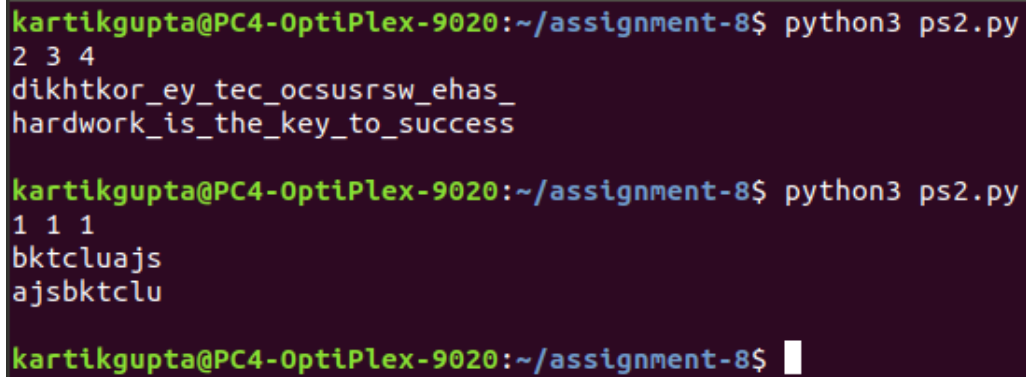
1 1 1

bktcluajs

Sample Output 1

ajsbktclu

2.7 Screenshots



```
kartikgupta@PC4-OptiPlex-9020:~/assignment-8$ python3 ps2.py
2 3 4
dikhtkor_ey_tec_ocsusrsw_ehas_
hardwork_is_the_key_to_success

kartikgupta@PC4-OptiPlex-9020:~/assignment-8$ python3 ps2.py
1 1 1
bktcluajs
ajsbktclu

kartikgupta@PC4-OptiPlex-9020:~/assignment-8$
```

Figure 4: Problem 2 Result

3 Appendix

3.1 Code for ps1.py

```
1
2 #
3 # @file ps1.py
4 #
5 # @author Kartik Gupta
6 # @date 27 September 2018
7 #
8
9 # Getting number of rows and columns
10 r,c = list([int(i) for i in input().split()])
11 listarr=[]
12 listnew=[]
13 x=1
14
15 # Initializing lists for the pattern
16 for i in range(106):
17     listarr.append('')
18
19 listnew=[]
20
21 # Getting characters for pattern
22 for i in range(r):
23     si=input()
24     l=list(si[:c])+[' ',' ',' ',' ',' ',' ',' ',' ',' '])
25     listarr.insert(i,l)
26
27
28 for i in range(r):
29     for j in range(c):
30         if listarr[i][j]=="S":
31             for k in range(1,c):
32                 if (listarr[i][j+k]=="S") and (listarr[i][j-k]=="S") :
33                     x+=2
34
35             listnew.append(x)
36             x=1
37
38 maxvalue=max(listnew)
39 minvalue=min(listnew)
40 print("Maximum matching pattern in horizontal direction" + " " + str(maxvalue)
41       )
42 print("Minimum matching pattern in horizontal direction" + " " + str(minvalue)
43       )
```

3.2 Code for ps2.py

```
1
2 #
3 # @file ps2.py
4 #
5 # @author Kartik Gupta
6 # @date 27 September 2018
7 #
8
9 # Creating groups for different characters
10 group1 = list("abcdefghi")
11 group2 = list("jklmnopqr")
12 group3 = list("stuvwxyz_")
13
14 # initializing list for groups
15 gr1=[]
16 gr2=[]
17 gr3=[]
18 gr1new=[]
19 gr2new=[]
20 gr3new=[]
21
22 # initializing list for index
23 index1=[]
24 index2=[]
25 index3=[]
26
27 #Getting k1,k2 and k3 values
28 k1,k2,k3 = [int(i) for i in input().split()]
29
30 # getting input string
31 word=input()
32 wordlist=list(word)
33
34 # Dividing the characters into different groups
35 for i in range(len(word)):
36     if wordlist[i] in group1:
37         gr1.append(wordlist[i])
38         index1.append(i)
39     elif wordlist[i] in group2:
40         gr2.append(wordlist[i])
41         index2.append(i)
42     elif wordlist[i] in group3:
43         gr3.append(wordlist[i])
44         index3.append(i)
45
46 # Rotating the groups
47 gr1new = (gr1[-k1:] + gr1[:-k1])
48 gr2new = (gr2[-k2:] + gr2[:-k2])
```

```

49 gr3new = (gr3[-k3:] + gr3[: -k3])
50
51 #Getting decrypted msg
52 p=q=r=0
53 for i in range(0,len(word)):
54     if i in index1:
55         wordlist[i]=gr1new[p]
56         p+=1
57     elif i in index2:
58         wordlist[i]=gr2new[q]
59         q+=1
60     elif i in index3:
61         wordlist[i]=gr3new[r]
62         r+=1
63
64 #displaying the decrypted msg
65 for i in wordlist[:]:
66     print (i, end = ' ')
67
68 print("\n")

```

References

- [1] GeeksforGeeks. *Pattern Matching*. <https://www.geeksforgeeks.org/program-to-print-hollow-rectangle-or-square-star-patterns/>.
- [2] StackExchange. *Shift Elements by n slices*. <https://codereview.stackexchange.com/questions/88684/shift-elements-left-by-n-indices-in-a-list>.