

Assignment-8

ELP - 718 Telecom Software Laboratory

**Shivaji Roy
2018JTM2002
2018-2020**

A report presented for the assignment-8



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

Contents

1	Objective	3
2	Problem Statement 1	3
2.1	Input and Output format	3
2.2	Test Cases	4
2.3	Screenshots	4
3	Problem Statement 2	4
3.1	Problem Statement	4
3.2	Input Output Format	5
3.3	Test Case	5
3.4	Difficulties/Issues Faced	5
3.5	Screenshots	6
4	Appendix	6
4.1	Appendix-A : code for ps1	6
4.2	Appendix-B : code for ps2	7

1 Objective

To develop our logical skills to solve the given problem with the help of basic python syntax.

2 Problem Statement 1

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.

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 .

Note: The two crosses cannot overlap, and the dimensions of each of the valid crosses should be maximal.

Constraints

- $2 \leq n \leq 105$
- $2 \leq m \leq 105$

2.1 Input and Output format

- **Input format**

The first line contains two space-separated integers, n and m. Each of the next lines n 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 jth character in the ith line is S, then (i,j) is a cell smart. Otherwise it's a dull cell..

- **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).

2.2 Test Cases

Sample Input1

```
5 6
SSSSSS
SDDDS
SSSSSS
SSDDSD
SSSSSS
```

Sample Output0

```
5 1
```

Sample Output1

```
6 6
DSDDSD
SSSSSS
DSDDSD
SSSSSS
DSDDSD
DSDDSD
```

Sample Input2

```
5 9
SSSSDSDDD
DDSDDDDDD
SSSSSDDDD
DDSDDSDDD
DSSSDDDDD
```

Sample Output2

```
9 1
```

2.3 Screenshots

3 Problem Statement 2

3.1 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.

3.2 Input Output Format

- **Input format**

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

- **Output format**

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

- **Constraints**

- $1 \leq \text{length of string} \leq 150$

- $1 \leq k_i \leq 150 (i = 1, 2, 3)$

3.3 Test Case

Sample Input: 2 3 4

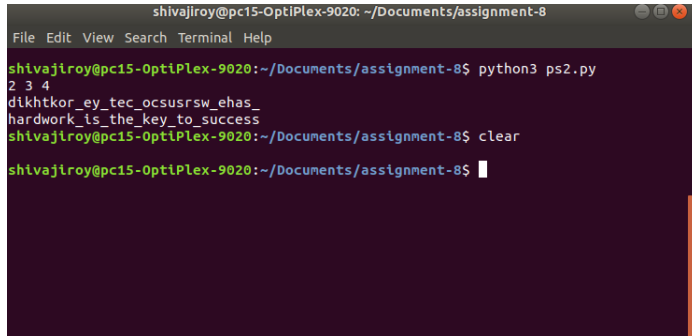
dikhtkor_ey_tec_ocsusrsw_ahas_

Sample Output: hardworkis_the_key_to_success

3.4 Difficulties/Issues Faced

- Error in code compiling.
- Error while giving inputs through Terminal.

3.5 Screenshots



A screenshot of a terminal window titled "shivajiroy@pc15-OptiPlex-9020: ~/Documents/assignment-8". The terminal shows the following commands and output:

```
File Edit View Search Terminal Help
shivajiroy@pc15-OptiPlex-9020:~/Documents/assignment-8$ python3 ps2.py
2 3 4
dikhtkor_ey_tec_ocsusrsw_ehas_
hardwork_is_the_key_to_success
shivajiroy@pc15-OptiPlex-9020:~/Documents/assignment-8$ clear
shivajiroy@pc15-OptiPlex-9020:~/Documents/assignment-8$
```

4 Appendix

4.1 Appendix-A : code for ps1

```
1 #include <stdio.h>
2 int main(void)
3 {
4     printf("Hello World!");
5 }
```

4.2 Appendix-B : code for ps2

```
1 ##### this is the second .py file #####
2
3 ##### write your code here #####
4 #rotate function
5 def rotate(lst,x):
6     copy = list(lst)
7     for i in range(len(lst)):
8         if x<0:
9             lst[i+x] = copy[i]
10        else:
11            lst[i] = copy[i-x]
12
13
14 #Create 3 groups
15 a1="abcdefghi"
16 a2="jklmnopqr"
17 a3="stuvwxyz_"
18
19 b1 =[]
20 b2 =[]
21 b3 =[]
22 i1=[]
23 i2=[]
24 i3=[]
25
26 #get key vakue from user
27 k1,k2,k3 = list(map(int,input().split()))
28
29 #get string
30 msg = input()
31 msg_list = list(msg)
32 #print(msg_list)
33
34 #now compair g1 in string and copy similaar char into s1
35 for i in range(0,len(msg)):
36     if msg_list[i] in a1:
37         b1.append(msg_list[i])
38         i1.append(i)
39
40     elif msg_list[i] in a2:
41         b2.append(msg_list[i])
42         i2.append(i)
43     elif msg_list[i] in a3:
44         b3.append(msg_list[i])
45         i3.append(i)
46
47
```

```

48
49 #rotate b1,b2,b3
50 rotate(b1,k1)
51 rotate(b2,k2)
52 rotate(b3,k3)
53
54
55
56 #get decrypted msg
57 p=q=r=0
58 for i in range(0,len(msg)+1):
59     if i in i1:
60         msg_list[i]=b1[p]
61         p+=1
62     elif i in index2:
63         msg_list[i]=b2[q]
64         q+=1
65     elif i in index3:
66         msg_list[i]=b3[r]
67         r+=1
68
69 #print(msg_list)
70
71 for i in msg_list[:]:
72     print(i, end = ' ')
73
74 print("")

```

References