Assignment-8

ELP - 718 Telecom Software Laboratory

Priyanshu Aggarwal

2018JTM2247

2018-2020

Assignment on Python Basics and Github

Bharti School Of

Telecommunication Technology and Management

IIT Delhi

India

September 27, 2018

Contents

List of Figures

1 Problem Statement-1

1.1 Problem Statement-1

IT 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.

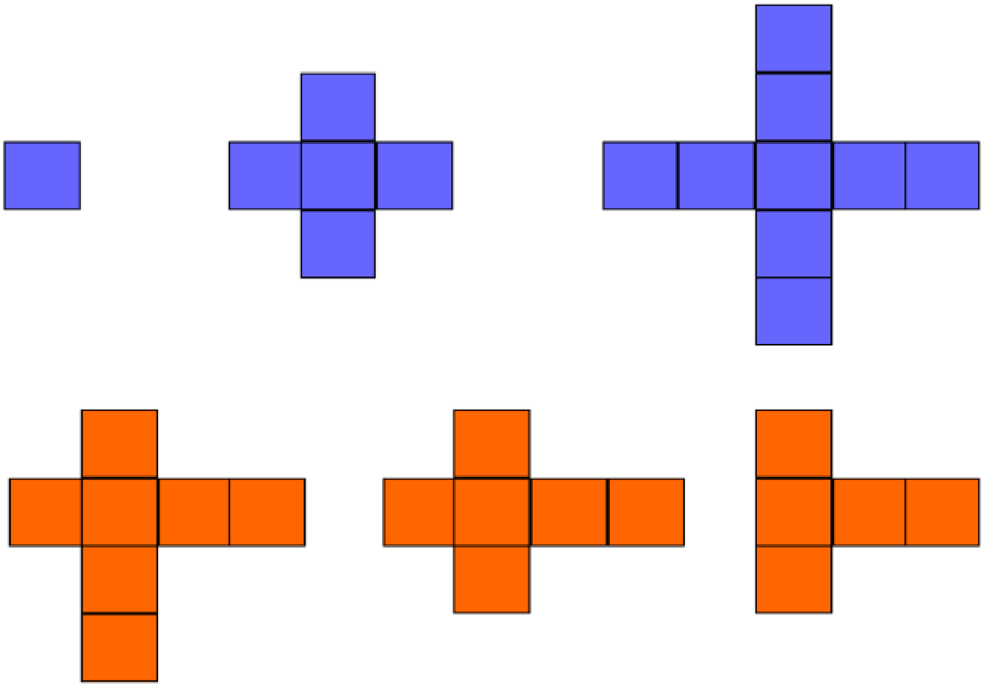


Figure 1: *In the diagram above, the blue crosses are valid and the orange ones are not valid.*

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 .

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

1.2 Assumptions

Constraints

- 2 <= n <= 105
- 2 <= m <= 105

1.3 Program Structure

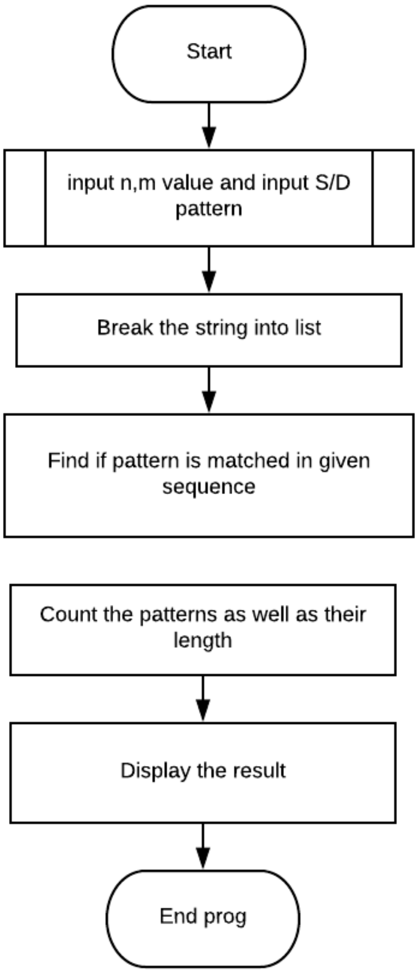

Start
input n,m value and input S/D pattern
Break the string into list
Find if pattern is matched in given sequence
Count the patterns as well as their length
Display the result
End prog

Figure 2: *Problem statement 1 flow chart*

1.4 Input and Output Format

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).

Sample Input 0

```
5 6
SSSSSS
SDDDSD
SSSSSS
SSDDSD
SSSSSS
```

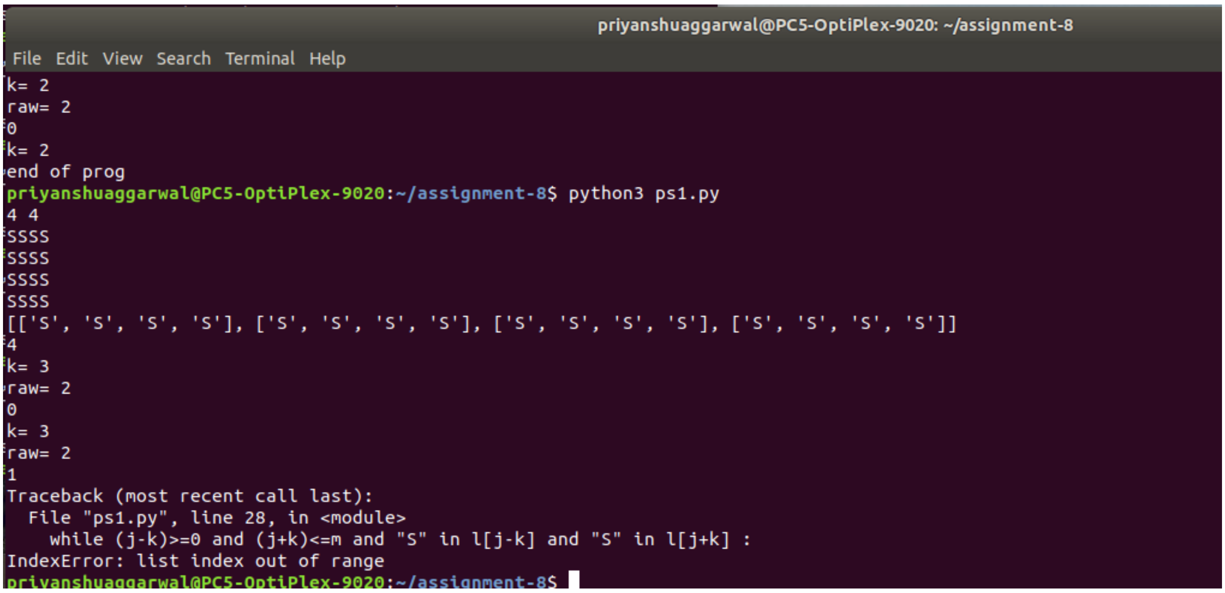
Sample Output

```
5 1
```

1.5 Difficulties

we are unable to get singel cahractert from input.

1.6 Screenshots


priyanshuaggarwal@PC5-OptiPlex-9020: ~/assignment-8
File Edit View Search Terminal Help

```
k= 2
raw= 2
0
k= 2
end of prog
priyanshuaggarwal@PC5-OptiPlex-9020:~/assignment-8$ python3 ps1.py
4 4
SSSS
SSSS
SSSS
SSSS
[['S', 'S', 'S', 'S'], ['S', 'S', 'S', 'S'], ['S', 'S', 'S', 'S'], ['S', 'S', 'S', 'S']]
4
k= 3
raw= 2
0
k= 3
raw= 2
1
Traceback (most recent call last):
  File "ps1.py", line 28, in <module>
    while (j-k)>=0 and (j+k)<=m and "S" in l[j-k] and "S" in l[j+k] :
IndexError: list index out of range
priyanshuaggarwal@PC5-OptiPlex-9020:~/assignment-8$
```

Figure 3: *Problem statement 1 output*

2 Problem Statement-2

2.1 Problem Statement-2

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, k1, k2, and k3. 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 ki positions in the message. Each group is rotated independently of the other two. Decrypting the message means doing a right rotation by ki positions within each group.

2.2 Assumptions

Constraints

- 1 <= Length of the string <=150
- 1<= ki <=150 (i=1,2,3)

2.3 Program Structure

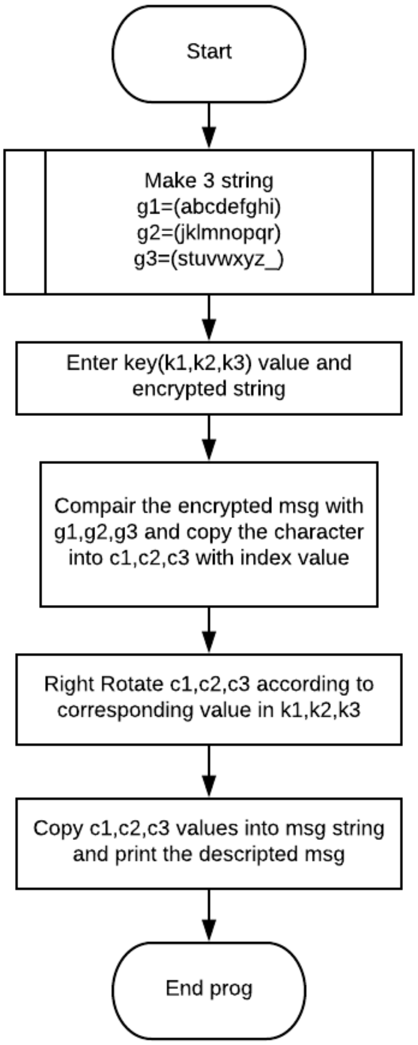

Start
Make 3 string
g1=(abcdefghi)
g2=(jklmnopqr)
g3=(stuvwxyz_)
Enter key(k1,k2,k3) value and encrypted string
Compair the encrypted msg with g1,g2,g3 and copy the character into c1,c2,c3 with index value
Right Rotate c1,c2,c3 according to corresponding value in k1,k2,k3
Copy c1,c2,c3 values into msg string and print the descripted msg
End prog

Figure 4: *Problem statement 2 flow chart*

2.4 Input and 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.

Sample Input 1

```
2 3 4
dikhtkor_ey_tec_ocsusrsw_ehas_
```

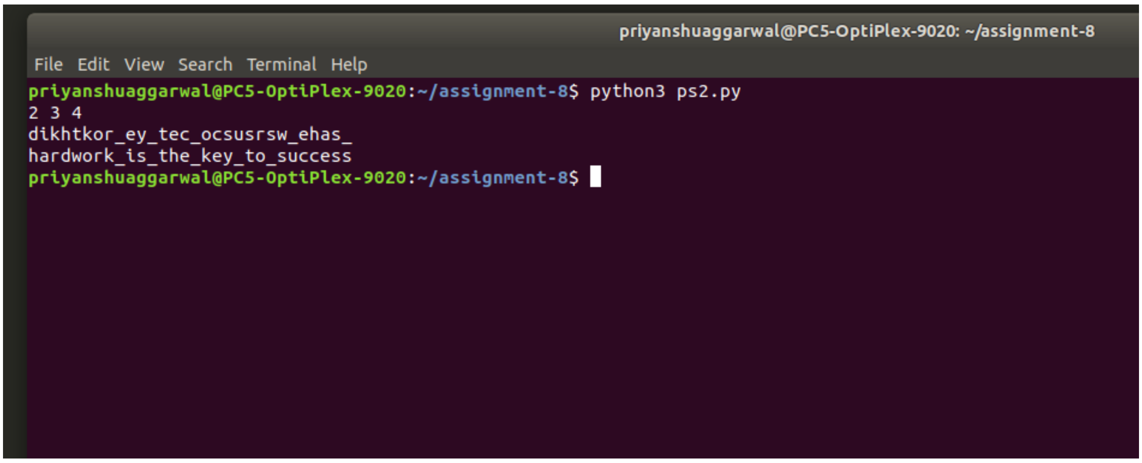
Sample Output 1

```
hardwork_is_the_key_to_success
```

2.5 Difficulties

Difficulties found to rotate the list.there is no inbuilt function.

2.6 Screenshots



priyanshuaggarwal@PC5-OptiPlex-9020: ~/assignment-8
File Edit View Search Terminal Help

```
priyanshuaggarwal@PC5-OptiPlex-9020:~/assignment-8$ python3 ps2.py
2 3 4
dikhtkor_ey_tec_ocsusrsw_ehas_
hardwork_is_the_key_to_success
priyanshuaggarwal@PC5-OptiPlex-9020:~/assignment-8$ 
```

Figure 5: *Problem statement 2 output*

3 Appendix

3.1 Code for ps1

```python
###### this is the first .py file ###########

####### write your code here ##########
n,m = list(map(int,input().split()))     #get n raw m coloum

l=[]

for i in range(0,n):
  t = list(input())
  if len(t) == m:
    l = l+ [t]
  else:
      print("Error")


print(l)
print(len(l))
raw_count=coloum_count=0
k=1
i=J=0
count=0

#chaeck pattern
for i in range(0,n):    #raw
  for j in range(0,m-1): #coloum value in the raw i
    if "S" in l[j]:
       #cheak horizontal pattern
      while (j-k)>=0 and (j+k)<=m and "S" in l[j-k] and "S" in l[j+k]
    :
            raw_count+=2
            print("raw=",raw_count)
            print(j-k)
            k+=1
        k=m-1
        print("k=",k)

              #check upper pattern
        #while l[i-k] in "S" and l[i+k] in "S" and (i-k)>=0 and (i+k)<=n:
        while (j-k)>=0 and (j+k)<=n and "S" in l[j-k] and "S" in l[j+k]:
            coloum_count+=2
            print("coloum=",coloum_count)
            k=k+k
        k=1

        if raw_count == coloum_count and raw_count != 0:
            count = raw_count + coloum_count + 1
            print("count=",count)

  #            raw_count = coloum_count = count = 0


        raw_count = coloum_count = count =0

print("end of prog")
```

]

3.2 Code for ps2

```
###### this is the first .py file ###########

####### write your code here ##########
n,m = list(map(int,input().split()))     #get n raw m coloum

l=[]

for i in range(0,n):
  t = list(input())
  if len(t) == m:
    l = l+ [t]
  else:
      print("Error")


print(l)
print(len(l))
raw_count=coloum_count=0
k=1
i=J=0
count=0

#chaeck pattern
for i in range(0,n):    #raw
  for j in range(0,m-1): #coloum value in the raw i
    if "S" in l[j]:
        #cheak horizontal pattern
      while (j-k)>=0 and (j+k)<=m and "S" in l[j-k] and "S" in l[j+k]:
        raw_count+=2
        print("raw=",raw_count)
```

```
            print(j-k)
            k+=1
        k=m-1
        print("k=",k)

                #check upper pattern
        #while l[i-k] in "S" and l[i+k] in "S" and (i-k)>=0 and (i+k)<=n:
        while (j-k)>=0 and (j+k)<=n and "S" in l[j-k] and "S" in l[j+k]:
            coloum_count+=2
            print("coloum=",coloum_count)
            k=k+k
        k=1

        if raw_count == coloum_count and raw_count != 0:
            count = raw_count + coloum_count + 1
            print("count=",count)

  #             raw_count = coloum_count = count = 0


        raw_count = coloum_count = count =0

print("end of prog")
```

]

References

[1] github.com. *github.comr*, 2018. https://github.com/2018jtm2247/assignment-8/blob/master/ps2.py.

[2] quora.com. *quora*, 2018. https://www.quora.com/How-do-I-take-integer-input-in-Python-delimited-by-space#.

[3] stackoverflow.com. *python*, 2018. https://stackoverflow.com/questions/2294493/how-to-get-the-position-of-a-character-in-python.

[4] tutorialspoint.com. *Python-basic*, 2018. https://www.tutorialspoint.com/python/string_find.htm.