# Assignment 8
# Python and Github

ELP-780

## Software Lab

Ravi Singh Thakur
2017EET2840

**Indian Institute Of Technology,
Delhi**

September 27, 2018

# Contents

# 1 Problem Statement 1

## 1.1 Problem Statement

**Find two largest crosses lengths of smart cell grid**

- Input a 2D array of strings consisting of DULL and SMART cells and find crosses which will be of odd lengths.

- Print largest two crosses found in matrix in non increasing order.

## 1.2 Assumptions

- Cells can either contains S or D character to represent SMART and DULL grid.

- Dimensions of 2D matrix can not be greater than $105 * 105$.

## 1.3 Algorithm Steps

- Take m and n as dimensions from user as input and store S or D in it.

- Replace S by 1 and D by 0 in this matrix.

- Take matrx $[-1, 0, 1]$ to determine cross.

- Multiply each cell value by this matrix rowwise and columnwise.

- If column and row sum is equal to 1 then sum it and print it.

- Repeat above steps for further calculations.

## 1.4 Input and Output Format

**I/O format**

- ### 1.4.1 Input Format

  - Two space separated values.
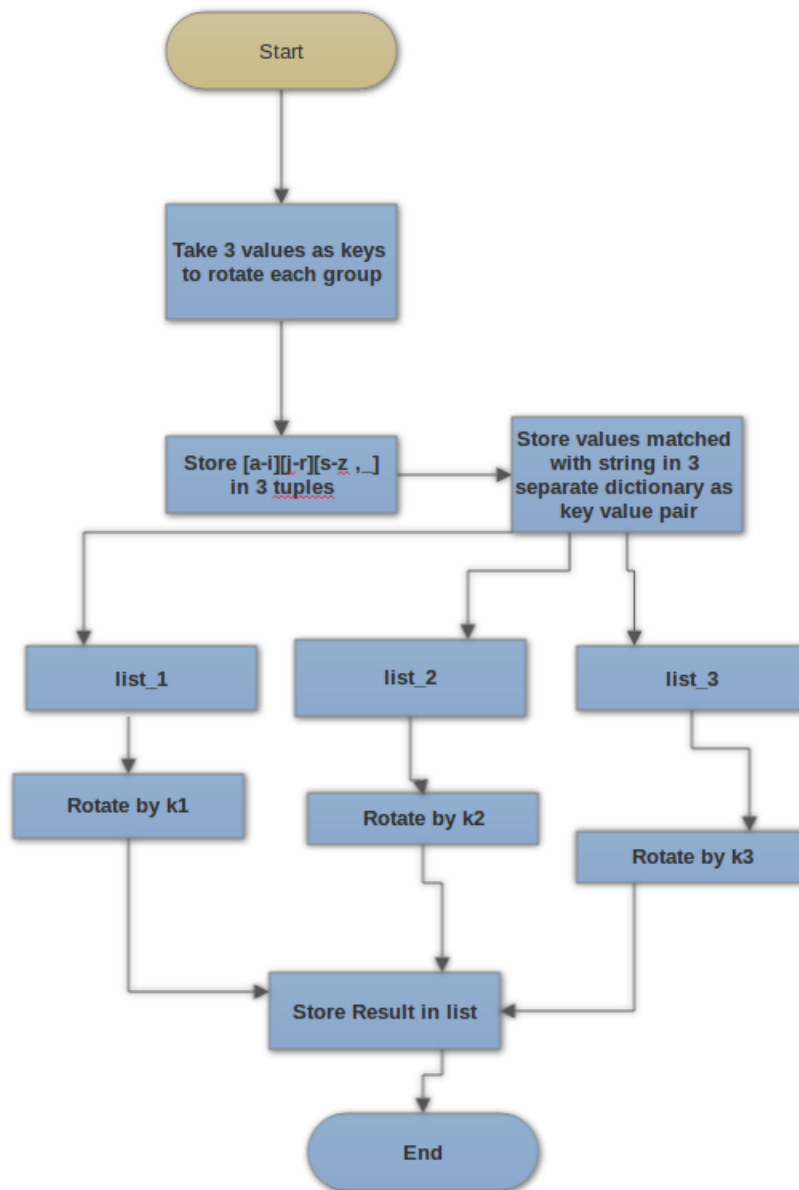  - Input taken as S or D for each cell.

- ### 1.4.2 Output Format

  - Two space separated values denoting sizes of two cross.

## 1.5 Difficulty faced

- Determining pattern for cross.

## 1.6 Program Structure

```
              ┌──────────────┐
              │    Start     │
              └──────────────┘
                     │
                     ▼
           ┌─────────────────────┐
           │  Take 3 values as keys │
           │  to rotate each group  │
           └─────────────────────┘
                     │
                     ▼
   ┌──────────────────┐      ┌──────────────────────┐
   │ Store [a-i][j-r][s-z ,_] │─────▶│ Store values matched │
   │     in 3 tuples      │      │  with string in 3    │
   └──────────────────┘      │ separate dictionary as │
                              │   key value pair     │
                              └──────────────────────┘
```

Start

Take 3 values as keys to rotate each group

Store [a-i][j-r][s-z ,_] in 3 tuples

Store values matched with string in 3 separate dictionary as key value pair

list_1

list_2

list_3

Rotate by k1

Rotate by k2

Rotate by k3

Store Result in list

End

## 1.7 Screenshots



```
ravi@emblab:~/Desktop/assignment-8$ python ps1.py
3 3
S
S
S
S
S
S
S
S
S
SSSSSSSSS
5 1
```

# 2  Problem Statement 2

## 2.1  Problem Statement

**Rotate keys and values of a string by k.**

- Divide alphabets and underscore into three groups and rotate by key and print result.

## 2.2  Assumptions

- Keys must be integer.

## 2.3  Algorithm Steps

- Alphabets and underscore is divided into three groups of tuple as required.

- Take keys as input for each group to rotate by.

- Rotate each dictionary by corresponding key.

- Store result in a list based on dictionary key values.

## 2.4  Input and Output Format

**I/O format**

- ### 2.4.1  Input Format

  - Three space separated keys,one for each group.
  - String is taken as input.
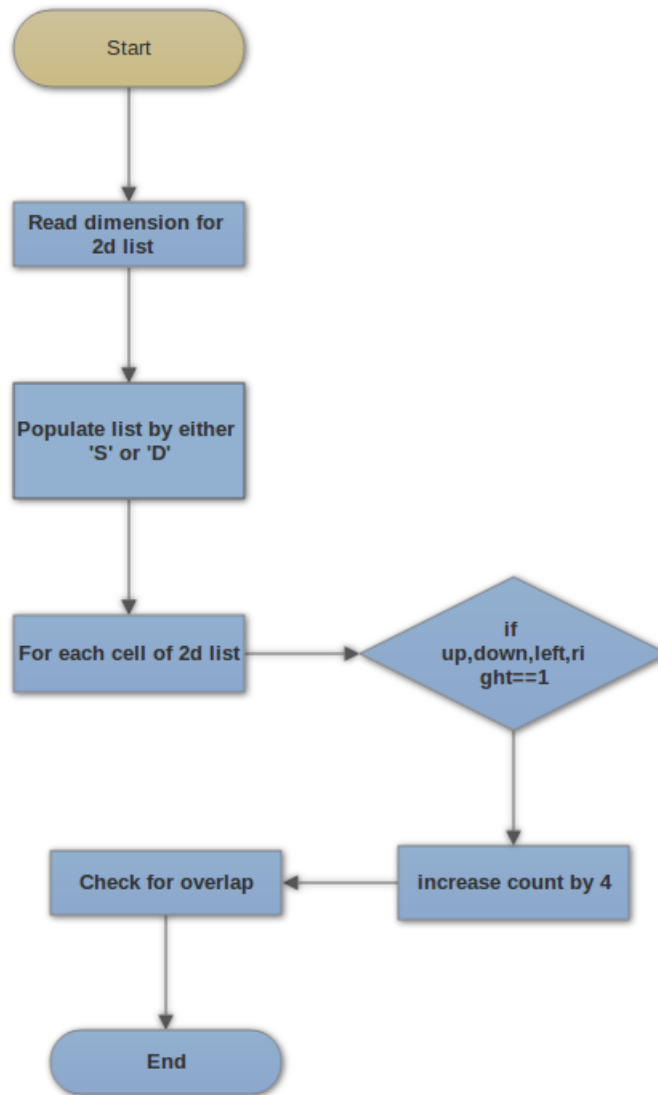
- ### 2.4.2  Output Format

  - Decrypted key as output string.

## 2.5  Difficulty faced

- rotate key value pair by key.

## 2.6 Program Structure

```
                    ┌─────────────┐
                    │    Start    │
                    └─────────────┘
                           │
                           ▼
                  ┌──────────────────┐
                  │ Read dimension for│
                  │     2d list      │
                  └──────────────────┘
                           │
                           ▼
                  ┌──────────────────┐
                  │ Populate list by │
                  │ either 'S' or 'D'│
                  └──────────────────┘
                           │
                           ▼
       ┌────────────────────┐        ┌──────────────┐
       │ For each cell of   │───────▶│      if      │
       │     2d list        │        │ up,down,left,│
       └────────────────────┘        │   right==1   │
                                      └──────────────┘
                                            │
                                            ▼
       ┌────────────────────┐        ┌──────────────────┐
       │ Check for overlap  │◀───────│ increase count by│
       └────────────────────┘        │        4         │
                │                     └──────────────────┘
                ▼
       ┌──────────────┐
       │     End      │
       └──────────────┘
```

## 2.7 Screenshots

```
ravi@emblab:~/Desktop/assignment-8$ python ps1.py
3 3
S
S
S
S
S
S
S
S
S
SSS
SSS
SSS
5 1
```

```
ravi@emblab:~/Desktop/assignment-8$ python ps1.py
3 4
S
S
S
S
S
S
S
S
S
S
S
S
SSSS
SSSS
SSSS
5 1
```

# 3 Appendix

**Appendix-A : code for ps1.py**

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

#importing array package from python library
from array import *
import copy

#This function checks for largest cross for each cell
def safe(a,x,y,m,n):
    count=1
    if a[x][y]=='D':
        return 0

    if m>3 and n>4:
        while x>0 and y>0 and x<m-1 and y<n-1 and a[x][y]=='S':
            if a[x+1][y]=='S' and a[x][y+1]=='S' and a[x-1][y]=='S' and a[x][y-1]=='S':
                x+=1
                y+=1
                count+=4
            else:
                a[x+1][y]=a[x][y+1]=a[x-1][y]=a[x][y-1]='D'
                break
    else:
        while x>0 and y>0 and x<m-1 and y<n-1 and a[x][y]=='S':
            if a[x+1][y]=='S' and a[x][y+1]=='S' and a[x-1][y]=='S' and a[x][y-1]=='S':
                a[x+1][y]=a[x][y+1]=a[x-1][y]=a[x][y-1]='D'
                x+=1
                y+=1
                count+=4

    return count


#taking input from user for dimensions of matrix
n,m = input().split()
n=int(n)
m=int(m)
#check boundary conditions
if n>150 or m>150:
    print("Out of range")
    print(exit)


arr = []
for i in range(n):
    arr.append([])
    for j in range(m):
        #appending either D or S in array
        arr[i].append(input())


#This loop is printing input taken in desired format

for r in arr:
    for c in r:
        print(c,end = "")
    print()


```

8

```
59  #Initializing list
60  lst=[]
61  #This loop checks for safe sequence for each cell.
62  for i in range(n):
63      count=1
64      for j in range(m):
65          count=safe(arr,i,j,n,m)
66          #append result in list
67          lst.append(count)
68
69  #Sort list in non increasing order
70  ans=sorted(lst,reverse=True)
71  #print result
72  print(ans[0],end=" ")
73  print(ans[1])
74
75
76
77
78
79  #initializing array to store values
80  '''
81  arr1=[[] for i in range(n)]
82  for j in range(m):
83      arr1[j].append(input())
84
85  arr = copy.deepcopy(arr1)
86
87  '''
```

### Appendix-B : code for ps2.py

```
1
2
3  #This function rotate the list by x number
4  def rotate(lst,x):
5      copy = list(lst)
6      for i in range(len(lst)):
7          if x<0:
8              lst[i+x] = copy[i]
9          else:
10             lst[i] = copy[i-x]
11
12 #Taking user input for 3 keys as k,l,m
13 k,l,m = input().split()
14 k=int(k)
15 l=int(l)
16 m=int(m)
17 #Taking input in str variable
18 str=input()
19
20 #Dividing all alphabets and _ in 3 groups in a list
21 t1=['a','b','c','d','e','f','g','h','i']
22 t2=['j','k','l','m','n','o','p','q','r']
23 t3=['s','t','u','v','w','x','y','z','_']
24
25 #initalizing list to store key value pair
26 a1=[]
27 a2=[]
28 a3=[]
29
30 #enumerate through each element of str and
31 #if it matches to corresponding group,add it to that list.
32 for i,j in enumerate(str,0):
33    if j in t1:
34      a1.append(j)
35
36    elif j in t2:
37      a2.append(j)
38
39    else:
40      a3.append(j)
41
42
43 #Calling function to rotate all 3 lists
44 rotate(a1,k)
45 rotate(a2,l)
46 rotate(a3,m)
47
48 #initalizing variables
49 g=0
50 res=[]
51 p,q,r=0,0,0
52 #enumerate through str to check
53 #which group this element belongs to
54 for i,j in enumerate(str,0):
55    if j in t1:
56      res.append(a1[p])
57      p+=1
58    elif j in t2:
59      res.append(a2[q])
60      q+=1
```

```
61      else :
62        res . append ( a3 [ r ] )
63        r+=1
64
65 #Printing  Result
66 for  c  in  res :
67    print ( c , end="" )
68 print ( )
```

# References

[1] https://stackoverflow.com/questions/33554620/rotating-values-in-a-list-python

[2] https://www.tutorialspoint.com/python/python2darray.htm