

In [2]:

```
def getConnectedComponents(arr,i,j,visited,options):
    #This is same as Rat in Maze
    if i<0 or j<0 or i>=len(arr) or j>=len(arr[0]) or arr[i][j] == 0 or visited[i][j]==True:
        return

    visited[i][j] = True

    #Now visit neighbours in 2D matrix arr
    if options == 1:
        getConnectedComponents(arr,i,j+1,visited,options) #RIGHT
        getConnectedComponents(arr,i,j-1,visited,options) #LEFT
    elif options == 2:
        getConnectedComponents(arr,i-1,j,visited,options) #TOP
        getConnectedComponents(arr,i+1,j,visited,options) #DOWN

    elif options == 3:
        getConnectedComponents(arr,i+1,j+1,visited,options) #Main Diagonal
        getConnectedComponents(arr,i+1,j-1,visited,options) #Reverse Diagonal

    elif options == 4:
        getConnectedComponents(arr,i-1,j,visited,options) #TOP
        getConnectedComponents(arr,i,j+1,visited,options) #RIGHT
        getConnectedComponents(arr,i,j-1,visited,options) #LEFT
        getConnectedComponents(arr,i+1,j,visited,options) #DOWN

    else:
        getConnectedComponents(arr,i-1,j,visited,options) #TOP
        getConnectedComponents(arr,i,j+1,visited,options) #RIGHT
        getConnectedComponents(arr,i+1,j+1,visited,options) #Main Diagonal
        getConnectedComponents(arr,i+1,j-1,visited,options) #Reverse Diagonal
        getConnectedComponents(arr,i,j-1,visited,options) #LEFT
        getConnectedComponents(arr,i+1,j,visited,options) #DOWN
```

In [3]:

```
print(" 1.Horizontal \n 2.Vertical \n 3.Diagonal \n 4.Horizontal_Vertical \n 5.Horizontal_V\n\n1.Horizontal\n2.Vertical\n3.Diagonal\n4.Horizontal_Vertical\n5.Horizontal_Vertical_Diagonal")
```

In [9]:

```

options = int(input())
m = int(input("Enter no of rows : ")) #Rows
n = int(input("Enter no of Columns : "))

#Cols
print("Enter the array ")
arr = [[int(input()) for x in range(n)] for y in range(m)]
visited = [[False]*n for _ in range(m)]
count = 0
for i in range(len(arr)):
    for j in range(len(arr[0])):
        #Visit cell/vertex only if it is land(0) and is not visited
        if arr[i][j] == 1 and visited[i][j]==False:
            getConnectedComponents(arr,i,j,visited,options)
            count+=1
if options in [3,5]:
    print('Components: ',count-2)
else:
    print('Components: ',count)

```

```

1
Enter no of rows : 5
Enter no of Columns : 5
Enter the array
1
0
1
0
1
0
1
0
1
0
1
0
1
1
1
0
0
1
0
1
0
1
0
0
Components:  10

```

In [3]:

```

options = int(input())

m = int(input("Enter no of rows : ")) #Rows
n = int(input("Enter no of Columns : "))

#Cols
print("Enter the array ")
arr = [[int(input()) for x in range(n)] for y in range(m)]
visited = [[False]*n for _ in range(m)]
count = 0
for i in range(len(arr)):
    for j in range(len(arr[0])):
        #Visit cell/vertex only if it is land(0) and is not visited
        if arr[i][j] == 1 and visited[i][j]==False:
            getConnectedComponents(arr,i,j,visited,options)
            count+=1
if options in [3,5]:
    print('Components: ',count-2)
else:
    print('Components: ',count)

```

```

2
Enter no of rows : 5
Enter no of Columns : 5
Enter the array
1
0
1
0
1
0
1
0
1
0
0
1
1
1
0
0
1
0
1
0
1
0
1
0
Components:  9

```

In [6]:

```

options = int(input())
m = int(input("Enter no of rows : ")) #Rows
n = int(input("Enter no of Columns : "))

#Cols
print("Enter the array ")
arr = [[int(input()) for x in range(n)] for y in range(m)]
visited = [[False]*n for _ in range(m)]
count = 0
for i in range(len(arr)):
    for j in range(len(arr[0])):
        #Visit cell/vertex only if it is land(0) and is not visited
        if arr[i][j] == 1 and visited[i][j]==False:
            getConnectedComponents(arr,i,j,visited,options)
            count+=1
if options in [3,5]:
    print('Components: ',count-2)
else:
    print('Components: ',count)

```

```

3
Enter no of rows : 5
Enter no of Columns : 5
Enter the array
1
0
1
0
1
0
1
0
1
0
0
1
1
1
0
0
1
0
1
0
1
0
1
0
1
0
1
0
Components:  3

```

In [7]:

```

options = int(input())
m = int(input("Enter no of rows : ")) #Rows
n = int(input("Enter no of Columns : "))

#Cols
print("Enter the array ")
arr = [[int(input()) for x in range(n)] for y in range(m)]
visited = [[False]*n for _ in range(m)]
count = 0
for i in range(len(arr)):
    for j in range(len(arr[0])):
        #Visit cell/vertex only if it is land(0) and is not visited
        if arr[i][j] == 1 and visited[i][j]==False:
            getConnectedComponents(arr,i,j,visited,options)
            count+=1
if options in [3,5]:
    print('Components: ',count-2)
else:
    print('Components: ',count)

```

```

4
Enter no of rows : 5
Enter no of Columns : 5
Enter the array
1
0
1
0
1
0
1
0
1
0
0
1
1
1
0
0
1
0
1
0
1
0
1
0
1
0
1
0
Components:  7

```

In [8]:

```

options = int(input())
m = int(input("Enter no of rows : ")) #Rows
n = int(input("Enter no of Columns : "))

#Cols
print("Enter the array ")
arr = [[int(input()) for x in range(n)] for y in range(m)]
visited = [[False]*n for _ in range(m)]
count = 0
for i in range(len(arr)):
    for j in range(len(arr[0])):
        #Visit cell/vertex only if it is land(0) and is not visited
        if arr[i][j] == 1 and visited[i][j]==False:
            getConnectedComponents(arr,i,j,visited,options)
            count+=1
if options in [3,5]:
    print('Components: ',count-2)
else:
    print('Components: ',count)

```

```

5
Enter no of rows : 5
Enter no of Columns : 5
Enter the array
1
0
1
0
1
0
1
0
1
0
0
1
1
1
0
0
1
0
1
0
1
0
1
0
1
0
1
0
Components:  1

```

In []:

