

```

'''
--> PROBLEM STATEMENT :
    Matrix = 3X3
    Sum = 15
    Digits = 1 to 9

--> APPROACH :
    ---> Combinations having sum 15
    ---> Unique Combinations having sum 15 (Unique Digits)
    ---> No. of combinations from each digits
    ---> Center element is the digit which have 4 combinations
    ---> Corner elements are the digits which have 3 combinations
    ---> Center elements in the corner row or column are the digits
which have 2 combinations

--> PATTERN :
    2  7  6

    9  5  1

    4  3  8

'''
# Given Data in Problem Statement
print()
digits = [1,2,3,4,5,6,7,8,9]
digits.sort()
num = 15

combinations = []
for i in range(len(digits)):
    temp_list1 = []
    for j in range(i+1,len(digits)):
        temp_list2 = []
        for k in range(j+1,len(digits)):
            if int(digits[i])+int(digits[j])+int(digits[k]) == 15:
                temp_list2.append(digits[i])
                temp_list2.append(digits[j])
                temp_list2.append(digits[k])
            combinations.append(temp_list2)
# print("Combinations : ",combinations)

# Remove empty lists using filter()
combinations = list(filter(None, combinations))
print("Unique Combinations (Sum=15): ",combinations)

```

```

# item at 0 index in the sublist is digit and the item at 1 index of
sublist is the count of that digit occurred in the unique combinations
digit_count_combination = []
for i in digits:
    temp_list = []
    count = 0
    temp_list.append(i)
    for j in combinations:
        if i in j:
            count += 1
    temp_list.append(count)
    digit_count_combination.append(temp_list)

# item at 0 index in the sublist is digit and the item at 1 index of
sublist is the count of that digit occurred in the unique combinations
print("Digits Occurance in Combinations : ",digit_count_combination)

# Sorting digit_count_combination list onn the basis of their digit's
count
sorted_count = sorted(digit_count_combination,key= lambda x:x[1])
print("Sorted Count : ",sorted_count)

```

```

# Center Digit : Digit occurred in 4 different combinations
# Corner Digits : Digit occurred in 3 different combinations
# Middle Digits : Digit occurred in 2 different combinations

```

```

# Finding digits with their position
center_digit = []
corner_digits = []
middle_digits = []

for i in sorted_count:
    if i[1] == 4:
        center_digit.append(i[0])
    elif(i[1]==3):
        corner_digits.append(i[0])
    else:
        middle_digits.append(i[0])

print()
print("Center Digit : ",center_digit)
print("Corner Digits : ",corner_digits)
print("Middle Digits : ",middle_digits)
middle_digits.reverse()

matrix = []
# print(matrix)

center_index = 0

```

```

corner_index = 0
middle_index = 0

for i in range(9):
    if i%2==0:
        if i==4:
            matrix.append(center_digit[center_index])
            center_index += 1
        else:
            matrix.append(corner_digits[corner_index])
            corner_index += 1
    else:
        matrix.append(middle_digits[middle_index])
        middle_index += 1

# Printing Matrix
print()
print("Matrix List : ",matrix)
print("Matrix : ")
matrix_index = 0
for i in range(3):
    for j in range(3):
        print(matrix[matrix_index],end=" ")
        matrix_index += 1
    print()

print("In this Matrix : Sum(rows_elements = Sum(columns_elements) =
Sum(diagonal_elements = 15))")
print()

```