def check\_correct\_row(k,i,n,matrix):

row\_sum = 0

for j in range(n):

if k != j:

row\_sum += abs(matrix[i][j])

if abs(matrix[i][k]) < row\_sum:

return False

return True

def make\_diagonally\_dominant(matrix,temp\_matrix):

n = len(matrix)

i=0

k=0

while i<n:

while k<n:

check = check\_correct\_row(i, k, n, matrix)

if check:

matrix[i], matrix[k] = matrix[k], matrix[i]

k+=1

k=i+1

i+=1

if matrix == temp\_matrix:

print(" system cannot be converted into diagonally dominant.")

else:

print("The new matrix is:")

for row in matrix:

print(row)

return matrix

def diagonally\_dominant\_matrix\_check(matrix):

n = len(matrix)

for i in range(n):

row\_sum = 0

for j in range(n):

if i != j:

row\_sum += abs(matrix[i][j])

if abs(matrix[i][i]) < row\_sum:

return False

return True

def input\_augmented\_matrix(n):

matrix =[]

print(f"Enter the augmented matrix:")

print(f"Each row must have {n + 1} elements.")

for i in range(n):

row = input(f"Row {i+1}: ").split()

if len(row) != n + 1:

print(f"Each row must have {n + 1} elements.")

return []

matrix.append([int(x) for x in row])

return matrix

def main():

n = int(input("Enter the size of the matrix (n): "))

augmented\_matrix = input\_augmented\_matrix(n)

if augmented\_matrix:

print("The augmented matrix is:")

for row in augmented\_matrix:

print(row)

diagonally\_dominant = diagonally\_dominant\_matrix\_check(augmented\_matrix)

temp\_matrix = [row[:] for row in augmented\_matrix]

if diagonally\_dominant:

print("The matrix is diagonally dominant.")

else:

print("The matrix is not diagonally dominant.")

new\_mat= make\_diagonally\_dominant(augmented\_matrix,temp\_matrix)

chek=diagonally\_dominant\_matrix\_check(new\_mat)

if n>1:

if not chek:

print("This matrix is not completely diagonally dominant as we can't do it.")

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Note:**

**I have checked the code on different inputs. You can also check it.**









