```python
from prettytable  import PrettyTable
from math import *

def make_table(n): #Create method table
    table = PrettyTable()
    fields = ["i"]
    for i in range(n):
        fields.append("x{0}".format(i+1))
    fields.append("Max &")
    table.field_names = fields
    return table

def calculate_new_Jacobi(x0,a,b):
    result = [0]*len(b)
    for i in range(0, len(a)):
        summ = 0
        for j in range(0, len(a)):
            if j != i:
                summ += a[i][j]*x0[j]
        result[i] = (b[i] - summ) / a[i][i]
    return result

def list_subtraction(a,b): #Substract the elements between two lists
    result = []
    for i in range(len(a)):
        result.append(abs(a[i] - b[i]))
    return result

def make_row(i,x0, dispersion):#Create a row who will be added to the table
    row = [i]
    for k in range(len(x0)):
            row.append(x0[k])
    row.append(dispersion)
    return row

def jacobi(a, b, x0, tol, niter):
    counter = 0
    dispersion = tol + 1
    x1 = []
    table = make_table(len(x0))
    table.add_row(make_row(counter, x0, "----"))
    while (dispersion > tol) and (counter < niter):
        x1 = calculate_new_Jacobi(x0, a, b)
        dispersion = max(list_subtraction(x1, x0))  # Norm
        x0 = x1
        counter += 1
        table.add_row(make_row(counter, x0, dispersion))
    if dispersion < tol:
        return table
    else:
        return "Failed"

def main():
    A = [[4,1,1,0,1],[-1,-3,1,1,0],[2,1,5,-1,-1],[-1,-1,-1,4,0],[0,2,-1,1,4]]
    b = [6,6,6,6,6]
    x0 = [0, 0, 0, 0, 0]
    print(jacobi(A, b, x0, 10**-7, 100))
main()
```