```python
def split_list(data):
    divide_index = data.index("")
    return data[:divide_index], data[divide_index+1:]


file = open("matrix.txt", "r")
data = file.read()
file.close()
data = data.split("\n")
matA, matB = split_list(data)
output = []
roundsA = 0
roundsB = 0
i = len(matA)
j = 0
k = 0


for index, x in enumerate(matA):
    matA[index] = list(map(int, matA[index].strip().split()))
    j = len(matA[index])
    roundsA += len(matA[index])
for index, x in enumerate(matB):
    matB[index] = list(map(int, matB[index].strip().split()))
    k = len(matB[index])  # paththata
    roundsB += len(matB[index])
if(j == len(matB)):
    print("Matrices can be multiplied")
    rounds = roundsB+roundsA+7
    output.append("MEM[0]=16'd5; //mat_a_base")
    output.append("MEM[1]=16'd{}; //mat_b_base".format(7+roundsA))
    output.append("MEM[2]=16'd{}; //mat_c_base \n".format(7+roundsA+roundsB))
    output.append("MEM[3]=16'd{}; //i_ref".format(i))
    output.append("MEM[4]=16'd{}; //j_ref \n".format(j))
    count = 5
    for i in matA:
        output.append("//{}".format(i))
        for j_item in i:
            output.append("MEM[{}]=16'd{};".format(count, j_item))
            count += 1
    output.append("\nMEM[{}]=16'd{}; //j_ref".format(count, j))
    output.append("MEM[{}]=16'd{}; //k_ref\n".format(count+1, k))
    count += 2
    for i in matB:
        output.append("//{}".format(i))
        for j_item in i:
            output.append("MEM[{}]=16'd{};".format(count, j_item))
            count += 1

    file = open("output.txt", "w")
    result = [[sum(a*b for a, b in zip(X_row, Y_col))
               for Y_col in zip(*matB)] for X_row in matA]
    output.append("\n//results should be..")
    for item in result:
        output.append("//{}".format(item))
    for element in output:
        file.write(element + "\n")
    file.close()
else:
    print("Matrices can't be multiplied")
```