Q5.

For this problem you need to do some operations on matrices.

Here are some explanations about the function in class:

a.matrix(int m, int n)

Initialize a matrix of size m * n.

b.void Add(matrix m1)

Add the original matrix by m1.

If two matrices cannot do this operation, then print "Not the same shape" .

c.void Sub(matrix m1)

Subtract the original matrix by m1.

If two matrices cannot do this operation, then print "Not the same shape" .

d.matrix Mul(matrix m1)

Multiply two matrices and print the result.

If two matrices cannot do this operation, then print "Cannot do multiplication!" .

e.void Inverse()

If inverse matrix exist, find and print inverse matrix of original matrix.

If not exist, then print "The Inverse Not Exist" .

(Hint: Use classical adjoint matrix to find inverse matrix.)

f. bool IsSameShape(matrix m1)

Return true if two matrices have same shape; otherwise return fasle.

g.bool SquareMatrix()

Return true if matrix is a square matrix; otherwise return fasle.

h.int Det(vector<vector<int> >& m);

Compute the determinant of matrix m.

See the template for the detail.

Input Format

Operator:

Enter the character "op" and do specific operations.

'+' : Add operation.

'-' : Subtract operation.

'*' : Multiply operation.

'd' : Compute the determinant.

'i' : Get Inverse matrix.

'q' : End program.

See more detail from Lab4.cpp.

Matrix:
1.Enter two integers to represent the shape of the matrix.
The first integer m represents the size of row, the second integer n represents the size of column.
2. Enter values sequentially from left to right and top to bottom.

See more detail from Sample input.

Output Format

You must output the result after doing the calculation.
If it is a valid operation, print the result.

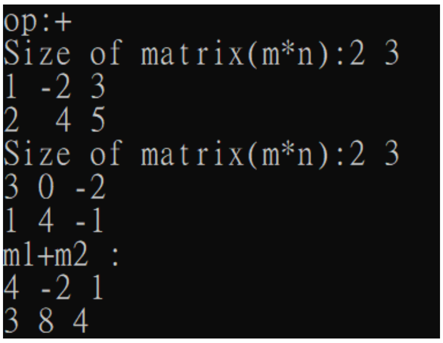
Note: Print out the inverse matrix with format “1/det(A) * adj(A)”, where det(A) represents determinant of A, adj(A) represents classical adjoint matrix of A.
(If you don't know what is a classic adjoint matrix, please google it.)

If it is invalid operation, then print out a specific message.

See more detail from Sample output.

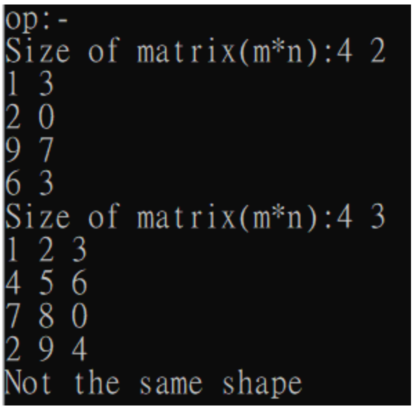
Sample Input & Output.

Ex1:



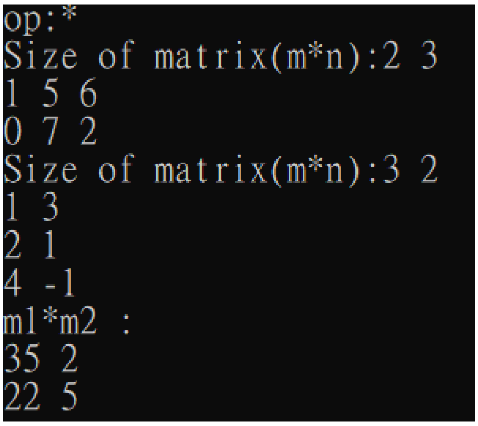
```
op:+
Size of matrix(m*n):2 3
1 -2 3
2  4 5
Size of matrix(m*n):2 3
3 0 -2
1 4 -1
m1+m2 :
4 -2 1
3 8 4
```

Ex2:



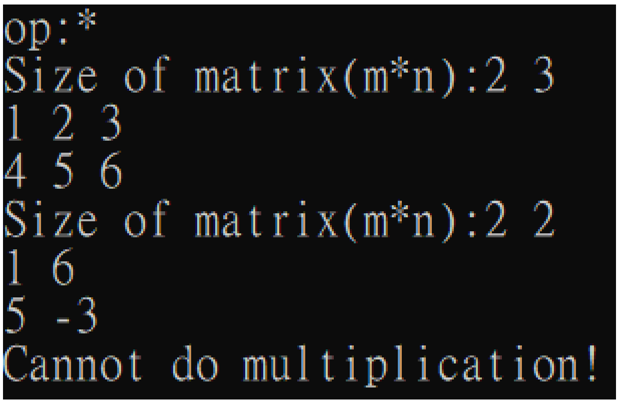
```
op:-
Size of matrix(m*n):4 2
1 3
2 0
9 7
6 3
Size of matrix(m*n):4 3
1 2 3
4 5 6
7 8 0
2 9 4
Not the same shape
```

Ex3:



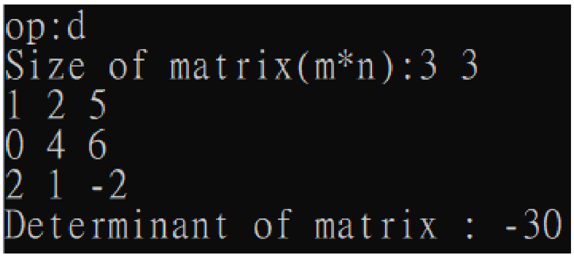
```
op:*
Size of matrix(m*n):2 3
1 5 6
0 7 2
Size of matrix(m*n):3 2
1 3
2 1
4 -1
m1*m2 :
35 2
22 5
```

Ex4:



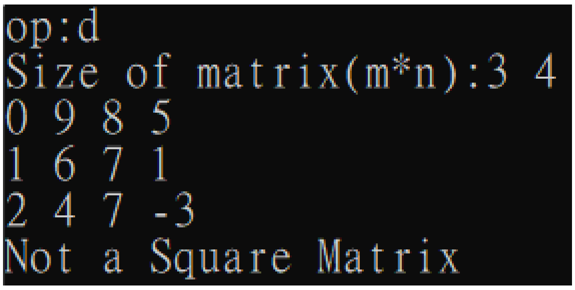
```
op:*
Size of matrix(m*n):2 3
1 2 3
4 5 6
Size of matrix(m*n):2 2
1 6
5 -3
Cannot do multiplication!
```

Ex5:



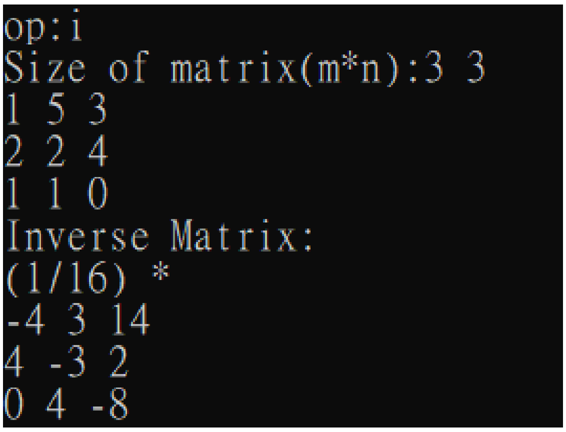
```
op:d
Size of matrix(m*n):3 3
1 2 5
0 4 6
2 1 -2
Determinant of matrix : -30
```

Ex6:



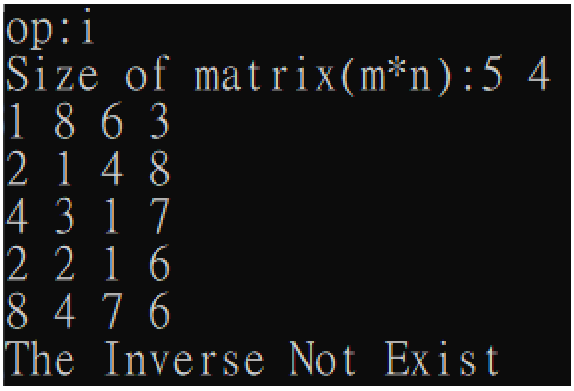
```
op:d
Size of matrix(m*n):3 4
0 9 8 5
1 6 7 1
2 4 7 -3
Not a Square Matrix
```

Ex7:



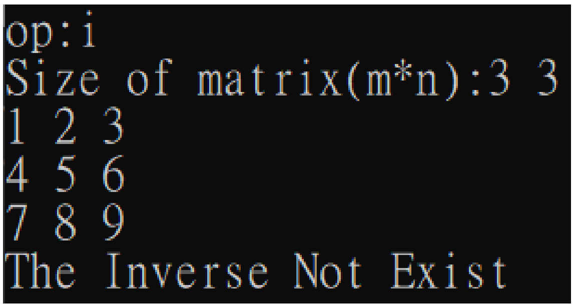
```
op:i
Size of matrix(m*n):3 3
1 5 3
2 2 4
1 1 0
Inverse Matrix:
(1/16) *
-4 3 14
4 -3 2
0 4 -8
```

Ex8:



```
op:i
Size of matrix(m*n):5 4
1 8 6 3
2 1 4 8
4 3 1 7
2 2 1 6
8 4 7 6
The Inverse Not Exist
```

Ex9:



```
op:i
Size of matrix(m*n):3 3
1 2 3
4 5 6
7 8 9
The Inverse Not Exist
```