Exercise (a)

```python
def matMul(A,B):
    if len(A[0]) == len(B): # cA == rB
        n = len(A)
        C = [[0 for nul in range(n)] for nul in range(n)]
        for i in range(n):
            for j in range(n):
                for k in range(n):
                    C[i][j] = C[i][j] + A[i][k] * B[k][j]
        return C
    return None

A = [[1, 2, 3],
     [4, 5, 6],
     [7, 8, 9]]
B = [[9, 8, 7],
     [6, 5, 4],
     [3, 2, 1]]

C = matMul(A,B)
print("Matrix C:")
for row in C:
    print(row)
# Output:
```

```
Matrix C:
[30, 24, 18]
[84, 69, 54]
[138, 114, 90]
```

Exercise (b)

```python
def matMul(A,B):
    rA, cA, rB, cB = len(A), len(A[0]), len(B), len(B[0])
    if cA == rB:
        C = [[0 for nul in range(cB)] for nul in range(rA)]
        for i in range(cA):
            for j in range(cB):
                for k in range(rB):
                    C[i][j] = C[i][j] + A[i][k] * B[k][j]
        return C
    return None

A = [[1, 2, 3],
     [4, 5, 6],
     [7, 8, 1]]
B = [[2, 8, 7, 4],
     [6, 5, 4, 4],
     [3, 2, 1, 4]]

C = matMul(A,B)
print("Matrix C: of Order rAxcB")
for row in C:
    print(row)

# Output:  Matrix C: of Order rAxcB
           [23, 24, 18, 24]
           [56, 69, 54, 60]
           [65, 98, 82, 64]
```

Exercise (c)

```python
import numpy as np

def matmul(A, B):
    if A.shape[1] != B.shape[0]:
        return "Error: Cannot multiply, dimensions not aligned!"
    return np.dot(A, B)

A = np.array([[1, 2],
              [3, 4],
              [5, 6]])

B = np.array([[7, 8, 9],
              [10, 11, 12]])
print("C:")
print(matmul(A, B))
```

✓ 0.0s

```
C:
[[ 27  30  33]
 [ 61  68  75]
 [ 95 106 117]]
```