

```
'''
Write a function in Python according to the given specification:
-----
```

Input parameters

A: Integer matrix (2D array with the same number of rows and columns)

Return value:

Return value is True if and only if
all values on both diagonals are equal.

Examples:

```
A = [
    [ 2, 1, 0, 1, 2 ]
    [ 3, 2, 1, 2, 3 ]
    [ 2, 3, 2, 3, 2 ]
    [ 3, 2, 1, 2, 3 ]
    [ 2, 1, 0, 1, 2 ]
]
```

Return value: True

```
A = [
    [ 8, 7, 8, 7 ]
    [ 8, 8, 8, 8 ]
    [ 8, 8, 8, 8 ]
    [ 8, 7, 8, 8 ]
]
```

Return value: False

```
'''
```

```
# Fill in necessary/missing statements.
# There may be one or more missing statements/commands/checks
# at each specified empty place.
```

```
def f( A ):
    N = len(a)
    value = A[0][0]
    if value != A[0][N-1]:
        return False
    _____ # fill in
    if value != A[j][j]: return False
    if value != A[j][N-1-j]: return False
    _____ # fill in
```

'''
Write a function in Python according to the given specification:

Input parameters

A: Integer matrix (2D array)

Return value:

Return value is equal to the number of columns
in the matrix which contain only 0's.

Examples:

```
A = [  
    [ 0, 1, 0, 0, 0 ]  
    [ 0, 0, 0, 0, 0 ]  
    [ 0, 0, 0, 3, 0 ]  
    [ 0, 0, 0, 0, 0 ]  
    [ 2, 1, 0, 0, 0 ]  
]
```

Return value: 2

```
A = [  
    [ 0, 0, 0 ]  
    [ 0, 0, 0 ]  
    [ 0, 0, 0 ]  
]
```

Return value: 3

```
'''  
  
def nZeroCols( a ):  
    n = 0  
    for j in range( len(a[0] )):  
        isOK = True  
        for i in range( len(a) ):  
            _____ # fill in missing statement(s)  
            _____  
  
    if isOK:  
        n += 1  
    return n
```

```
'''
Write a function in Python according to the given specification:
-----
```

Input parameters

A: integer matrix (2D array)

Return value:

Return value is True if and only if the maximum value in the matrix appears only once in the matrix. Otherwise, the return value is False.

Examples:

```
A = [
    [ 1, 2, 3, 4, 5 ]
    [ 5, 4, 3, 2, 1 ]
    [ 6, 5, 4, 3, 2 ]
    [ 3, 4, 5, 6, 0 ]
    [ 1, 1, 2, 1, 1 ]
]
```

Return value: False

```
A = [
    [ 2, 3, 4, 5 ]
    [ 1, 0, 1, 0 ]
    [ 1, 9, 1, 1 ]
    [ 8, 1, 1, 0 ]
]
```

Return value: True

```
'''
```

```
def f( A ):
    maxval = max( A[0] )
    for row in A:
        maxval = max( maxval, max( row ) )
    counter = 0
    for row in A:
        for j in range( len(row) ):
            _____ # fill in missing code
            _____
        if _____
            return False
    return True
```

'''
Write a function in Python according to the given specification:

Input parameters

A: integer matrix (2D array)

Return value:

Return value is True if and only if the values in each column
appear in ascending order from top to bottom.
Otherwise, the return value is False.

Examples:

```
A = [  
    [ 1, 2, 1, 0 ]  
    [ 2, 4, 3, 2 ]  
    [ 3, 6, 5, 7 ]  
    [ 4, 8, 7, 9 ]  
    ]
```

Return value: True

```
A = [  
    [ 1, 1, 1, 5, 1 ]  
    [ 2, 2, 2, 4, 2 ]  
    [ 3, 3, 3, 3, 3 ]  
    [ 4, 4, 4, 2, 4 ]  
    [ 5, 5, 5, 1, 6 ]  
    ]
```

Return value: False

'''

```
def isAscend( A ):  
    for j in range( len(A[0]) ):  
        columnOK = True  
        for i in range( 1, len(a) ):  
            if _____: # fill missing statements  
                _____  
                _____  
        if columnOK == False:  
            return False  
    return True
```

'''
Write a function in Python according to the given specification:

Input parameters

A: Square integer matrix (2D array with same number of rows and columns, all of equal length)
K: Integer

Return value:

The function returns true if and only if both conditions hold:
All values which are not on the main diagonal of A are smaller than K
All values on the main diagonal are bigger than K+1.
Otherwise the function returns false.

Examples:

```
A = [  
    [ 7, 1, 0, 1, 2 ]  
    [ 3, 8, 1, 2, 3 ]  
    [ 2, 3, 9, 3, 2 ]  
    [ 3, 2, 1, 6, 3 ]  
    [ 2, 1, 0, 1, 6 ]  
]
```

K = 4

Return value: True

```
A = [  
    [ 10, 7, 8, 7 ]  
    [ 8, 9, 8, 8 ]  
    [ 8, 8, 9, 8 ]  
    [ 8, 7, 8, 9 ]  
]
```

K = 9

Return value: False

'''

```
def myFunc( A, K ):  
    for i in range( len(A) ):  
        for j in range( len(A[i] )):  
            if i != j and _____ # fill in missing lines  
                return _____  
            if i == j and _____  
                return False  
    return _____
```

```
'''
Write a function in Python according to the given specification:
-----
```

Input parameters

A: 2D rectangular integer array (the length of all rows is the same)
K: Integer

Return value:

The function returns true if and only if
the number of columns in which there are no duplicate values
is equal to K.
Otherwise the function returns false.

Examples:

```
A = [
    [ 7, 4, 0, 1, 2 ]
    [ 6, 2, 1, 1, 3 ]
    [ 5, 3, 9, 1, 0 ]
    [ 3, 2, 2, 2, 1 ]
    [ 4, 4, 4, 1, 4 ]
]
```

K = 3

Return value: True (Columns 0, 2 and 4 do not contain any duplicate values,
column 1 contains duplicate value 4 and 2
and column 3 contains duplicate value 1

```
A = [
    [ 0, 1, 1, 7 ]
    [ 8, 1, 2, 8 ]
    [ 8, 1, 3, 8 ]
    [ 8, 7, 1, 9 ]
]
```

K = 1

Return value: False (There is no column without a duplicate value)

```
'''
```

```
def myFunc( A, K ):
    goodColumns = 0
    for j in range( len(A[0]) ):
        column = []
        columnOK = True
        for i in range( len(A) ):
            if A[i][j] in column:
                _____ # fill in missing statements
            else:
                _____
        if columnOK:
            goodColumns += 1

    return _____
```

...
Write a function in Python according to the given specification:

Input parameters

A: 2D rectangular integer array (the length of all rows is the same)

Return value:

The function returns true if and only if exactly one column in the array contains only negative values and the other columns contain only positive values. Otherwise the function returns false.

Examples:

```
A = [
    [ 7, 4, 3, -1, 2 ]
    [ 6, 2, 1, -1, 3 ]
    [ 5, 3, 9, -1, 3 ]
    [ 3, 2, 2, -2, 1 ]
    [ 4, 4, 4, -1, 4 ]
]
```

Return value: True (The fourth column contains only negative values,
other columns contain only positive values.)

```
A = [
    [ 4, -1, 1, 7 ]
    [ 8, -1, 2, 8 ]
    [ 8, -1, 3, -8 ]
    [ 8, -7, 1, 9 ]
]
```

Return value: False (The last column does not contain only positive values.)

```
A = [
    [ 4, 1, 1, 7 ]
    [ 8, 1, -2, 8 ]
    [ 8, 1, -3, 8 ]
    [ 8, 7, 1, 9 ]
]
```

Return value: False (The third column contains both positive and negative values.)

...

```
def F( A ):
    negatives = [x for x in A[0] if x < 0 ]
    if len( negatives ) != 1:
        return False
    jColumn = A[0].index( negatives[0] )
    for i in range( len(A) ):
        if _____
            return _____
    for i in range( len(A) ):
        for j in range( len(A[i] ) ):
            if _____
                return _____
    return True
```

'''
Write a function in Python according to the given specification:

Input parameters

A: Square integer matrix (2D array with same number of rows and columns, all of equal length)

Return value:

The function returns true if and only if each column contains exactly one minimum value and the minimum value lies on the matrix diagonal. Otherwise the function returns false.

Examples:

```
A = [
    [ 1, 24, 3, 0, 18 ]
    [ 27, 5, 3, 0, 28 ]
    [ 35, 23, 2, 0, 18 ]
    [ 43, 11, 3, -6, 28 ]
    [ 14, 8, 3, 12, 4 ]
]
```

Return value: True

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

Return value: False (The minimums in the 1st and in the 2nd column are not unique.)

```
A = [
    [ 4, 1, 1, 7 ]
    [ 8, 0, 3, 8 ]
    [ 8, 1, 2, 8 ]
    [ 8, 7, 5, 2 ]
]
```

Return value: False (The minimum in the 3rd column is not on the main diagonal.)

'''

```
def F( A ):
    for j in range( len(A[0] )):
        minval = A[0][j]
        minpos = 0
        for i in range( 1, len(A) ):
            if A[i][j] > minval:
                continue
            if A[i][j] < minval:
                _____ # fill in the missing statements
            if A[i][j] == minval:
                return False
        if minpos != j:
            return _____
    return _____
```