Matrix

Two dimensional Array

- A list which all elements have a specific type is called array.
- We can define a multidimensional list.
- A one dimensional array is called vector
- A two dimensional array is called matrix

-
  ```
   M = [ [1, 2, 3],            # A 3 × 3 matrix, as nested lists
         [4, 5, 6],            # Code can span lines if bracketed
         [7, 8, 9] ]
  ```
- print(M)
- **[[1, 2, 3], [4, 5, 6], [7, 8, 9]]**
- Access by two index: M[i][j]
- print(M[1][2])

Table of Student Grades

Imagine a class of 7 students that have a quiz every week for 5 weeks. The instructor records the grades in a table. A particular cell of the table is designated by student number and week number. For example:

- The grade for student 0 week 1 is 42
- The grade for student 3 week 4 is 93
- The grade for student 6 week 2 is 78

Student	Week				
	0	1	2	3	4
0	99	42	74	83	100
1	90	91	72	88	95
2	88	61	74	89	96
3	61	89	82	98	93
4	93	73	75	78	99
5	50	65	92	87	94
6	43	98	78	56	99

gradeTable

A compact notation for specifying a cell uses the **row** and **column** indexes like this:

gradeTable[row][col]

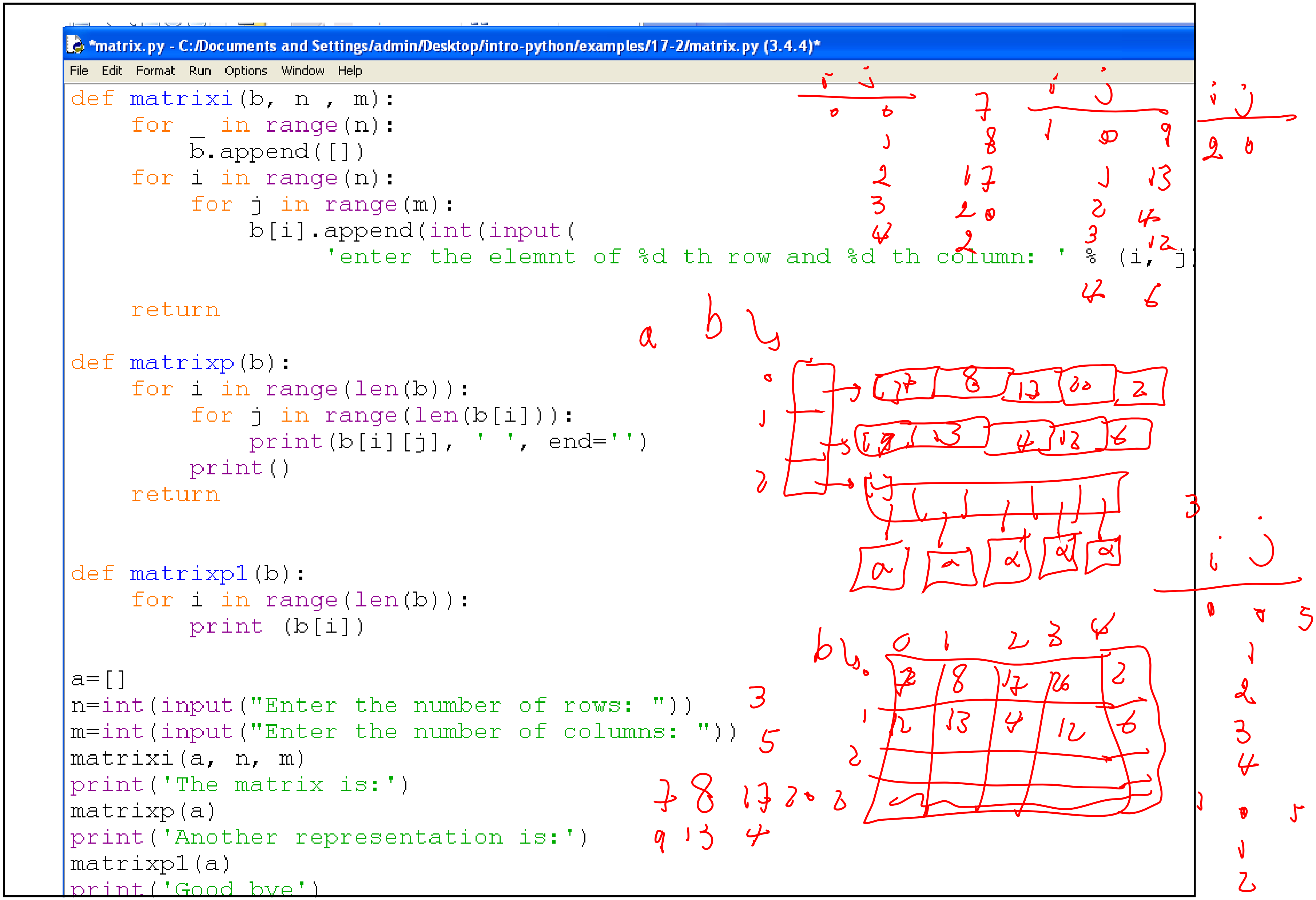
For example:

- gradeTable[0][1] is 42
- gradeTable[3][4] is 93
- gradeTable[6][2] is 78

۰ ۱ ۲ ... ۳۰

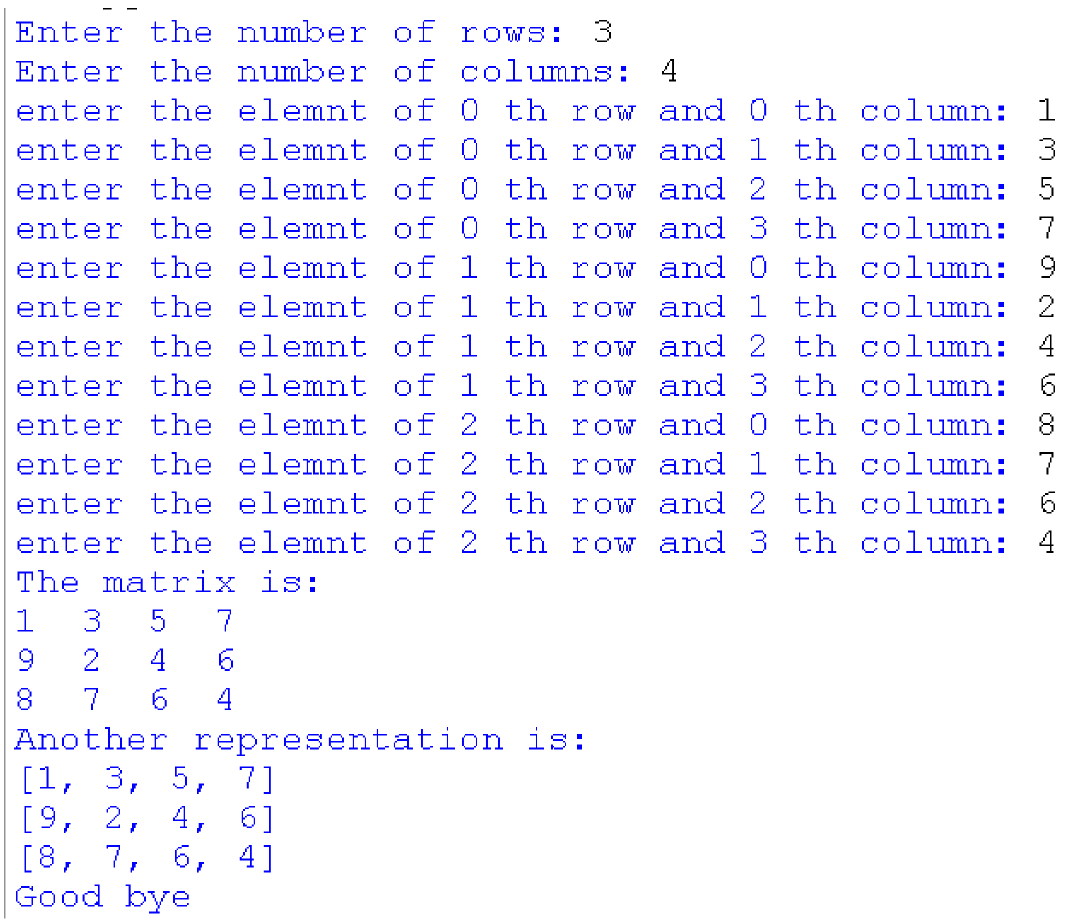
دروس ۰ α α
ردیف ۱
۲
:
افراد ۱۱

Read and write matrix



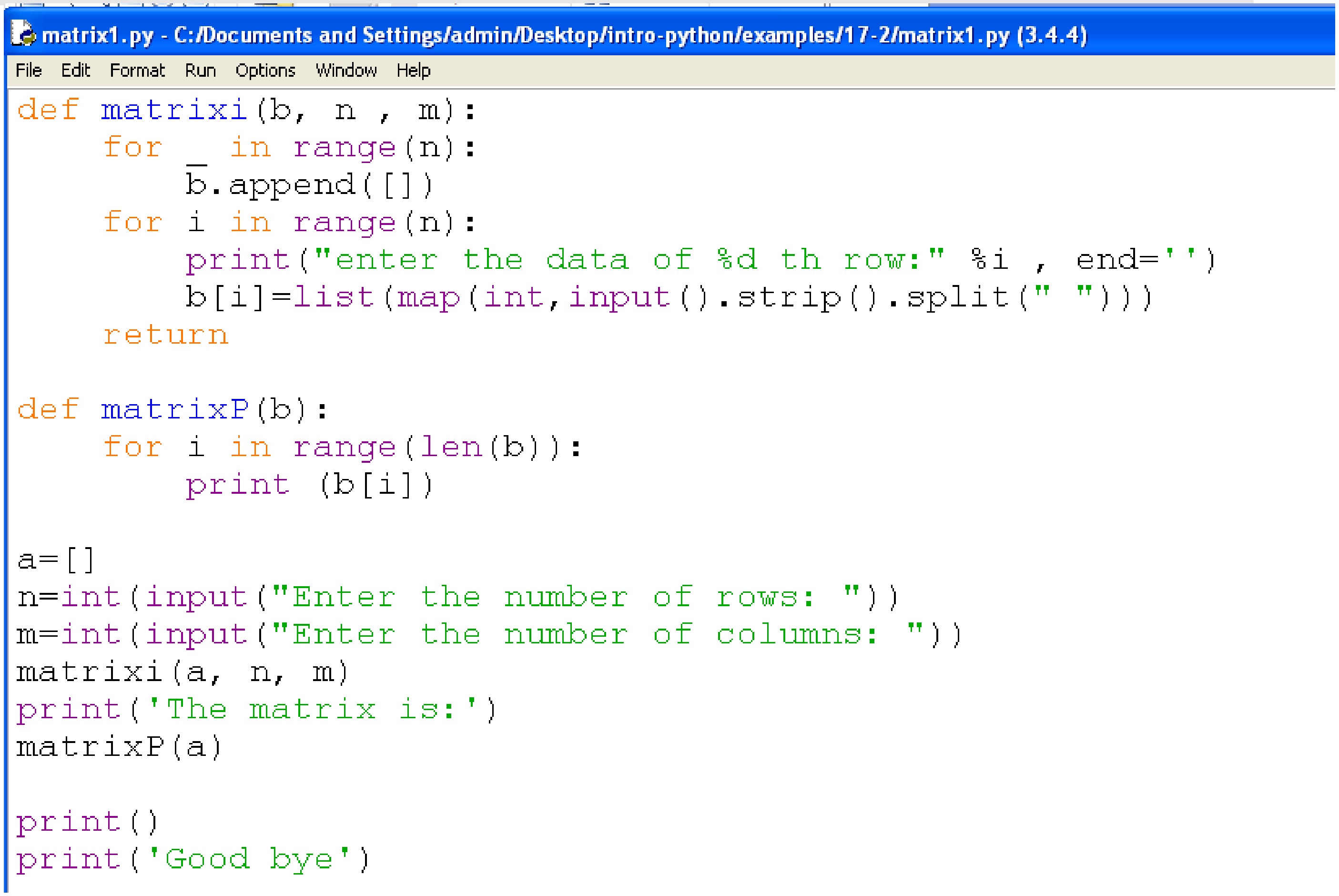
```
def matrixi(b, n , m):
    for _ in range(n):
        b.append([])
    for i in range(n):
        for j in range(m):
            b[i].append(int(input(
                'enter the elemnt of %d th row and %d th column: ' % (i, j)

    return

def matrixp(b):
    for i in range(len(b)):
        for j in range(len(b[i])):
            print(b[i][j], ' ', end='')
        print()
    return


def matrixp1(b):
    for i in range(len(b)):
        print (b[i])

a=[]
n=int(input("Enter the number of rows: "))
m=int(input("Enter the number of columns: "))
matrixi(a, n, m)
print('The matrix is:')
matrixp(a)
print('Another representation is:')
matrixp1(a)
print('Good bye')
```

Read and write matrix



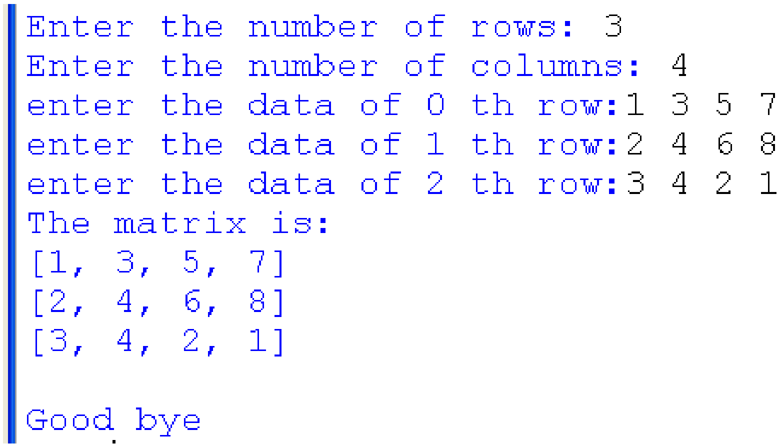
```
Enter the number of rows: 3
Enter the number of columns: 4
enter the elemnt of 0 th row and 0 th column: 1
enter the elemnt of 0 th row and 1 th column: 3
enter the elemnt of 0 th row and 2 th column: 5
enter the elemnt of 0 th row and 3 th column: 7
enter the elemnt of 1 th row and 0 th column: 9
enter the elemnt of 1 th row and 1 th column: 2
enter the elemnt of 1 th row and 2 th column: 4
enter the elemnt of 1 th row and 3 th column: 6
enter the elemnt of 2 th row and 0 th column: 8
enter the elemnt of 2 th row and 1 th column: 7
enter the elemnt of 2 th row and 2 th column: 6
enter the elemnt of 2 th row and 3 th column: 4
The matrix is:
1  3  5  7
9  2  4  6
8  7  6  4
Another representation is:
[1, 3, 5, 7]
[9, 2, 4, 6]
[8, 7, 6, 4]
Good bye
```

Read and write matrix


matrix1.py - C:/Documents and Settings/admin/Desktop/intro-python/examples/17-2/matrix1.py (3.4.4)
File Edit Format Run Options Window Help

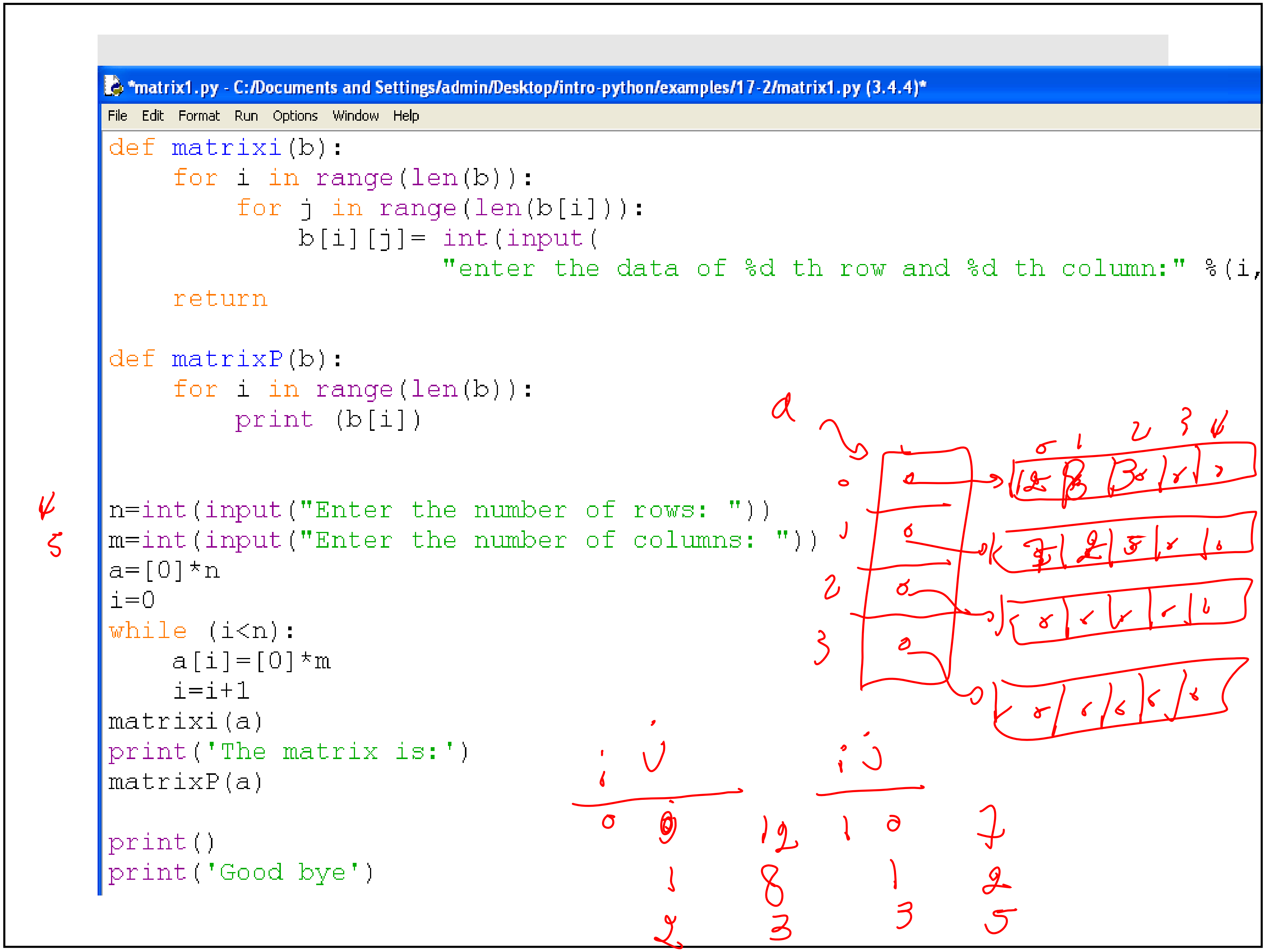
```
def matrixi(b, n , m):
    for _ in range(n):
        b.append([])
    for i in range(n):
        print("enter the data of %d th row:" %i , end='')
        b[i]=list(map(int,input().strip().split(" ")))
    return

def matrixP(b):
    for i in range(len(b)):
        print (b[i])

a=[]
n=int(input("Enter the number of rows: "))
m=int(input("Enter the number of columns: "))
matrixi(a, n, m)
print('The matrix is:')
matrixP(a)

print()
print('Good bye')
```

Read and write matrix



```
Enter the number of rows: 3
Enter the number of columns: 4
enter the data of 0 th row:1 3 5 7
enter the data of 1 th row:2 4 6 8
enter the data of 2 th row:3 4 2 1
The matrix is:
[1, 3, 5, 7]
[2, 4, 6, 8]
[3, 4, 2, 1]

Good bye
```

$$\begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 1 & 3 & 5 & 7 \\ 1 & 2 & 4 & 6 & 8 \\ 2 & 3 & 4 & 2 & 1 \end{array}$$



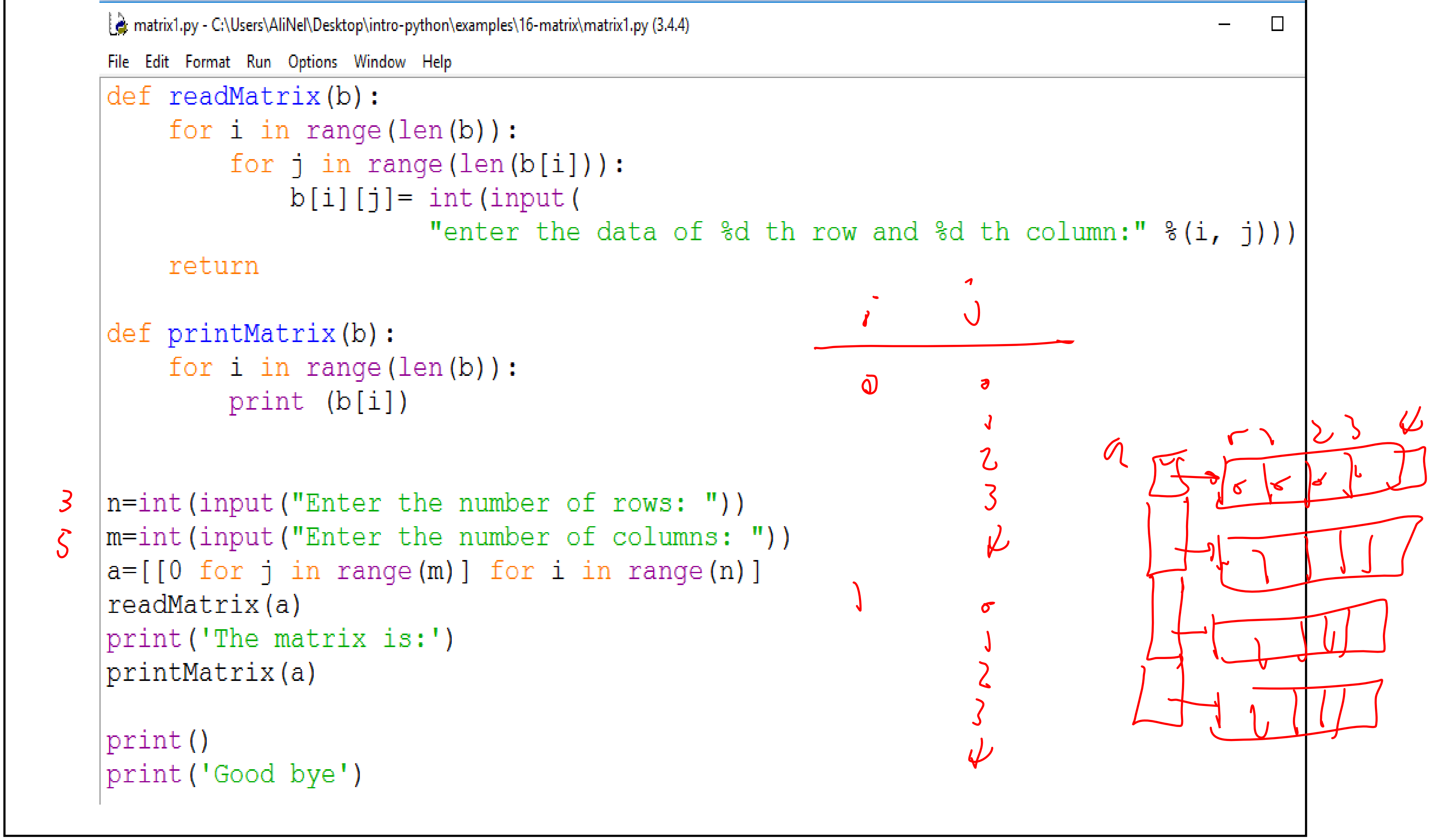
matrix1.py - C:/Documents and Settings/admin/Desktop/intro-python/examples/17-2/matrix1.py (3.4.4)
File Edit Format Run Options Window Help
4
5
a
0 1 2 3 4
0
1
2
3
i j
0 0
1
2
12
8
3
i j
1 0
1
3
7
9
5

```python
def matrixi(b):
    for i in range(len(b)):
        for j in range(len(b[i])):
            b[i][j]= int(input(
                 "enter the data of %d th row and %d th column:" %(i,
    return

def matrixP(b):
    for i in range(len(b)):
        print (b[i])


n=int(input("Enter the number of rows: "))
m=int(input("Enter the number of columns: "))
a=[0]*n
i=0
while (i<n):
    a[i]=[0]*m
    i=i+1
matrixi(a)
print('The matrix is:')
matrixP(a)

print()
print('Good bye')
```

Comprehensions

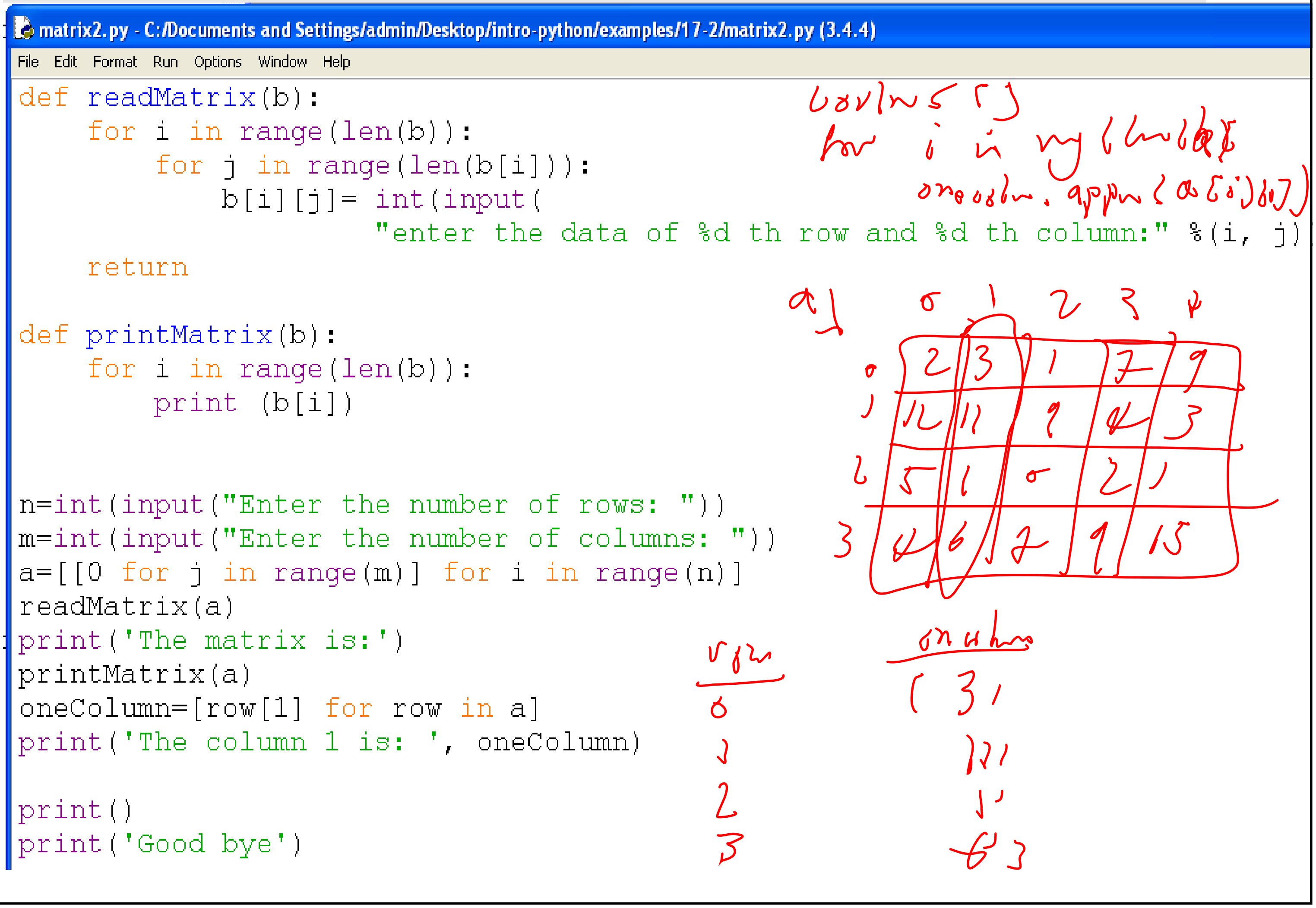
- In addition to sequence operations and list methods, Python includes a more advanced operation known as a *list comprehension expression, which turns out to be a powerful* way to process structures like our matrix. Suppose, for instance, that we need to extract the second column of our sample matrix. It’s easy to grab rows by simple indexing because the matrix is stored by rows, but it’s almost as easy to get a column with a list comprehension

Read and write matrix


matrix1.py - C:\Users\AliNel\Desktop\intro-python\examples\16-matrix\matrix1.py (3.4.4)
File Edit Format Run Options Window Help
3
5
i j
0 0 1 2 3 4
1 0 1 2 3 4
a
0 1 2 3 4
0 0 0 0

```
def readMatrix(b):
    for i in range(len(b)):
        for j in range(len(b[i])):
            b[i][j]= int(input(
                     "enter the data of %d th row and %d th column:" %(i, j)))
    return

def printMatrix(b):
    for i in range(len(b)):
        print (b[i])


n=int(input("Enter the number of rows: "))
m=int(input("Enter the number of columns: "))
a=[[0 for j in range(m)] for i in range(n)]
readMatrix(a)
print('The matrix is:')
printMatrix(a)

print()
print('Good bye')
```

Comprehensions


matrix2.py - C:/Documents and Settings/admin/Desktop/intro-python/examples/17-2/matrix2.py (3.4.4)
File Edit Format Run Options Window Help

```
def readMatrix(b):
    for i in range(len(b)):
        for j in range(len(b[i])):
            b[i][j]= int(input(
                     "enter the data of %d th row and %d th column:" %(i, j))
    return

def printMatrix(b):
    for i in range(len(b)):
        print (b[i])


n=int(input("Enter the number of rows: "))
m=int(input("Enter the number of columns: "))
a=[[0 for j in range(m)] for i in range(n)]
readMatrix(a)
print('The matrix is:')
printMatrix(a)
oneColumn=[row[1] for row in a]
print('The column 1 is: ', oneColumn)

print()
print('Good bye')
```

```
oneColumn = []
for i in range(len(a)):
    oneColumn.append(a[i][1])
```

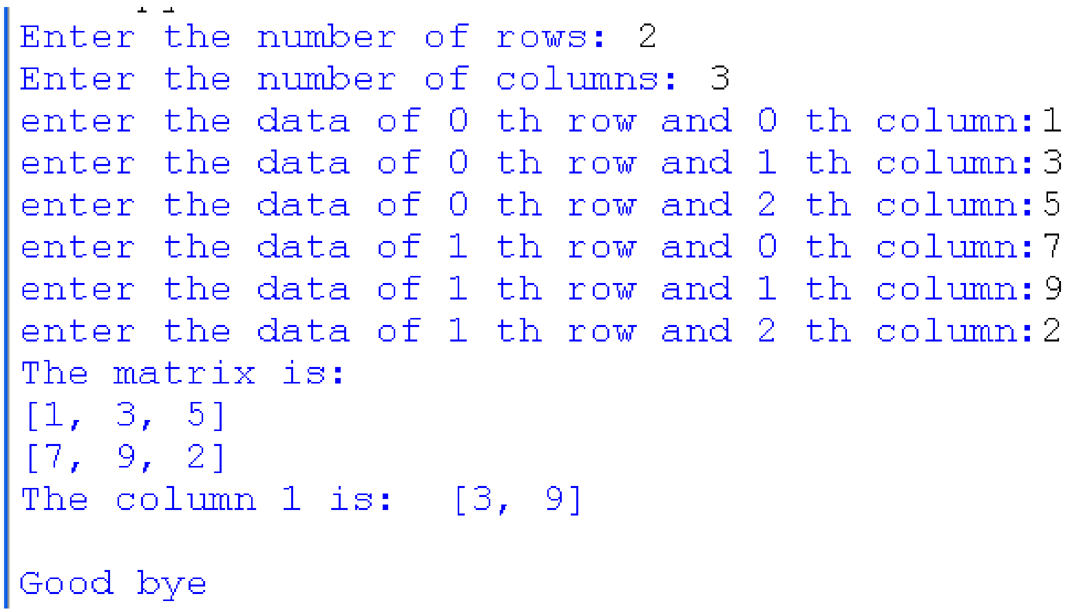
a

$$\begin{array}{c|ccccc} & 0 & 1 & 2 & 3 & 4 \\ \hline 0 & 2 & 3 & 1 & 7 & 9 \\ 1 & 12 & 11 & 9 & 4 & 3 \\ 2 & 5 & 1 & 0 & 2 & 1 \\ 3 & 4 & 6 & 7 & 9 & 15 \end{array}$$

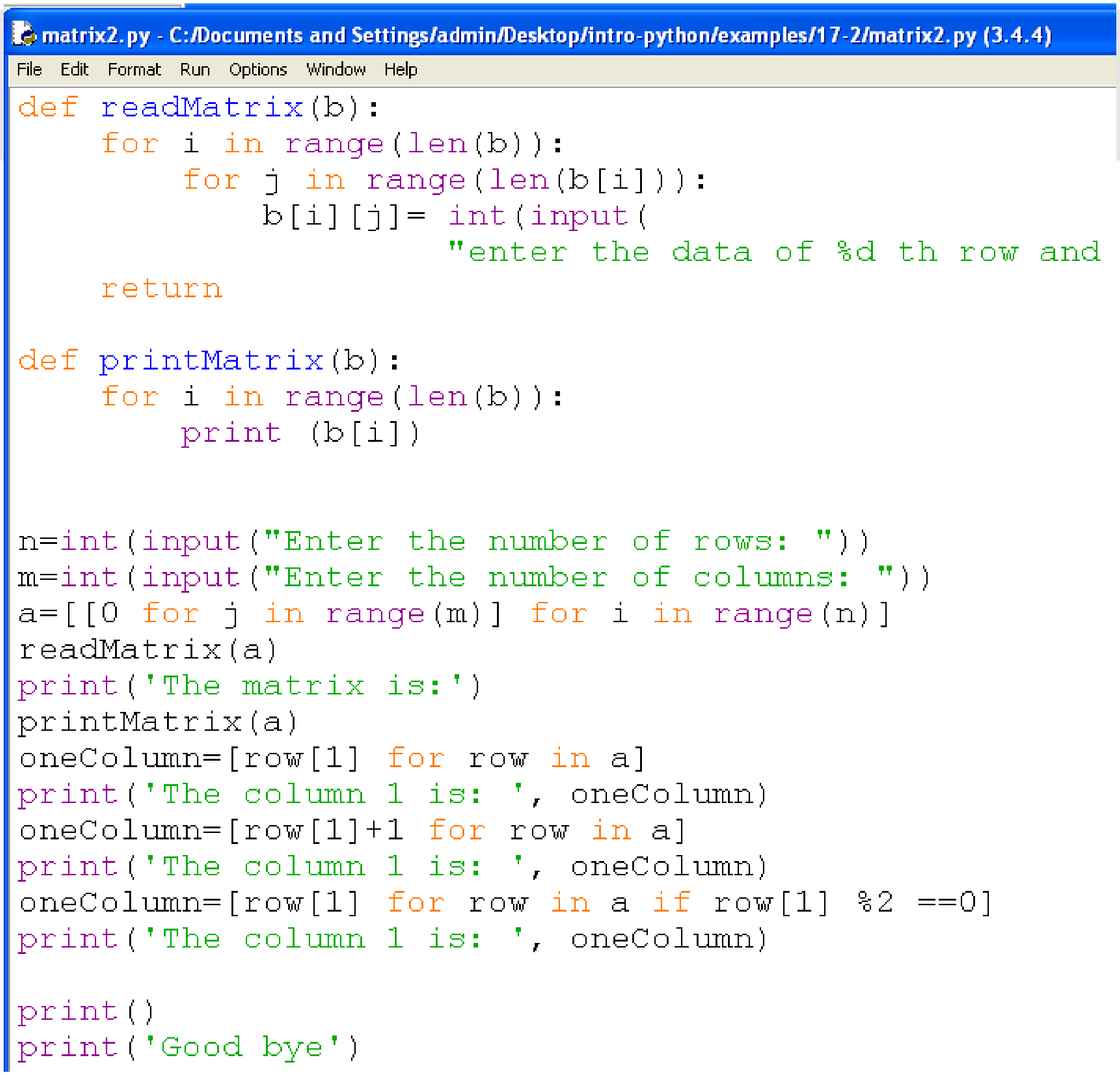
rows
0
1
2
3

oneColumn
[3,
11,
1,
6]

Comprehensions

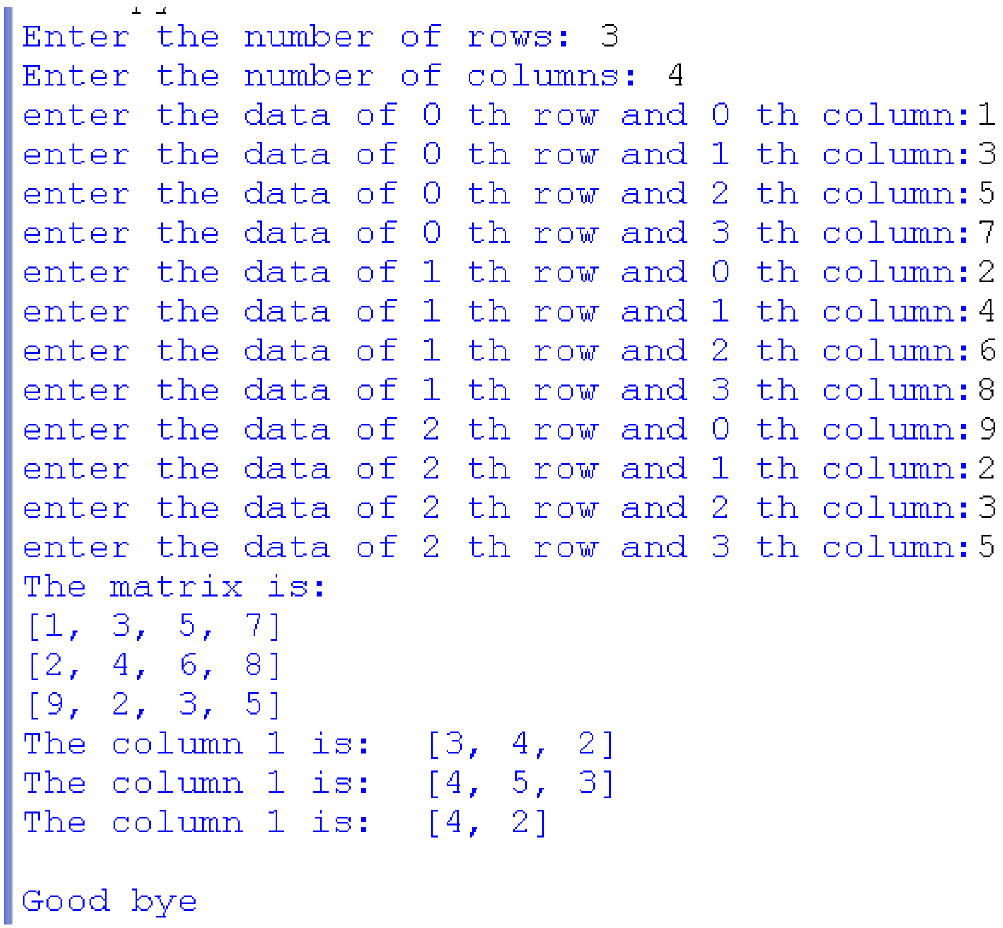


```
Enter the number of rows: 2
Enter the number of columns: 3
enter the data of 0 th row and 0 th column:1
enter the data of 0 th row and 1 th column:3
enter the data of 0 th row and 2 th column:5
enter the data of 1 th row and 0 th column:7
enter the data of 1 th row and 1 th column:9
enter the data of 1 th row and 2 th column:2
The matrix is:
[1, 3, 5]
[7, 9, 2]
The column 1 is:  [3, 9]

Good bye
```


matrix2.py - C:/Documents and Settings/admin/Desktop/intro-python/examples/17-2/matrix2.py (3.4.4)
File
Edit
Format
Run
Options
Window
Help

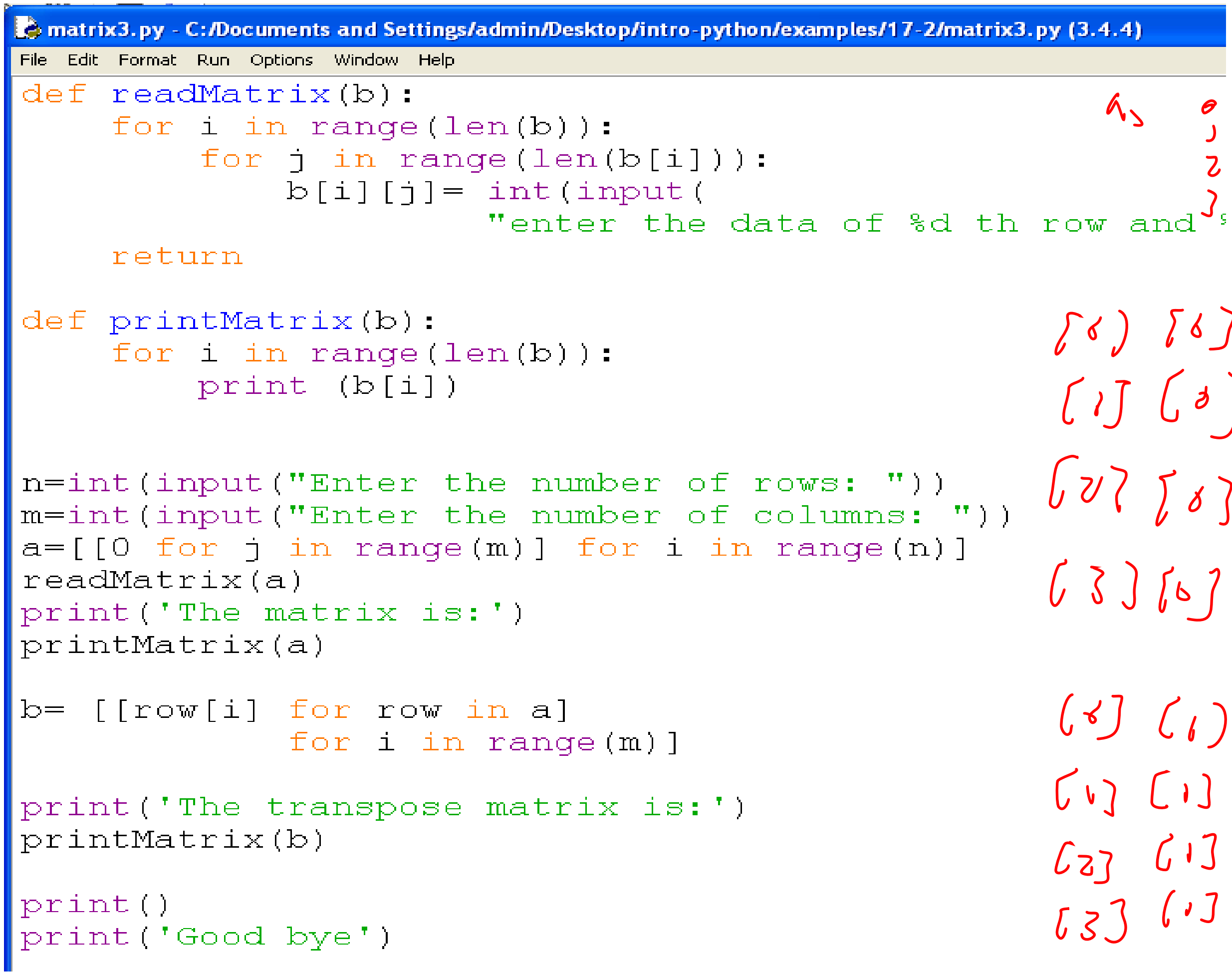
```
def readMatrix(b):
    for i in range(len(b)):
        for j in range(len(b[i])):
            b[i][j]= int(input(
                     "enter the data of %d th row and
    return

def printMatrix(b):
    for i in range(len(b)):
        print (b[i])


n=int(input("Enter the number of rows: "))
m=int(input("Enter the number of columns: "))
a=[[0 for j in range(m)] for i in range(n)]
readMatrix(a)
print('The matrix is:')
printMatrix(a)
oneColumn=[row[1] for row in a]
print('The column 1 is: ', oneColumn)
oneColumn=[row[1]+1 for row in a]
print('The column 1 is: ', oneColumn)
oneColumn=[row[1] for row in a if row[1] %2 ==0]
print('The column 1 is: ', oneColumn)

print()
print('Good bye')
```

Comprehensions



```
Enter the number of rows: 3
Enter the number of columns: 4
enter the data of 0 th row and 0 th column:1
enter the data of 0 th row and 1 th column:3
enter the data of 0 th row and 2 th column:5
enter the data of 0 th row and 3 th column:7
enter the data of 1 th row and 0 th column:2
enter the data of 1 th row and 1 th column:4
enter the data of 1 th row and 2 th column:6
enter the data of 1 th row and 3 th column:8
enter the data of 2 th row and 0 th column:9
enter the data of 2 th row and 1 th column:2
enter the data of 2 th row and 2 th column:3
enter the data of 2 th row and 3 th column:5
The matrix is:
[1, 3, 5, 7]
[2, 4, 6, 8]
[9, 2, 3, 5]
The column 1 is:  [3, 4, 2]
The column 1 is:  [4, 5, 3]
The column 1 is:  [4, 2]

Good bye
```

Matrix transpose



matrix3.py - C:/Documents and Settings/admin/Desktop/intro-python/examples/17-2/matrix3.py (3.4.4)
File Edit Format Run Options Window Help

```
def readMatrix(b):
    for i in range(len(b)):
        for j in range(len(b[i])):
            b[i][j]= int(input(
                         "enter the data of %d th row and
    return

def printMatrix(b):
    for i in range(len(b)):
        print (b[i])


n=int(input("Enter the number of rows: "))
m=int(input("Enter the number of columns: "))
a=[[0 for j in range(m)] for i in range(n)]
readMatrix(a)
print('The matrix is:')
printMatrix(a)

b= [[row[i] for row in a]
            for i in range(m)]

print('The transpose matrix is:')
printMatrix(b)

print()
print('Good bye')
```

0 1 2 3

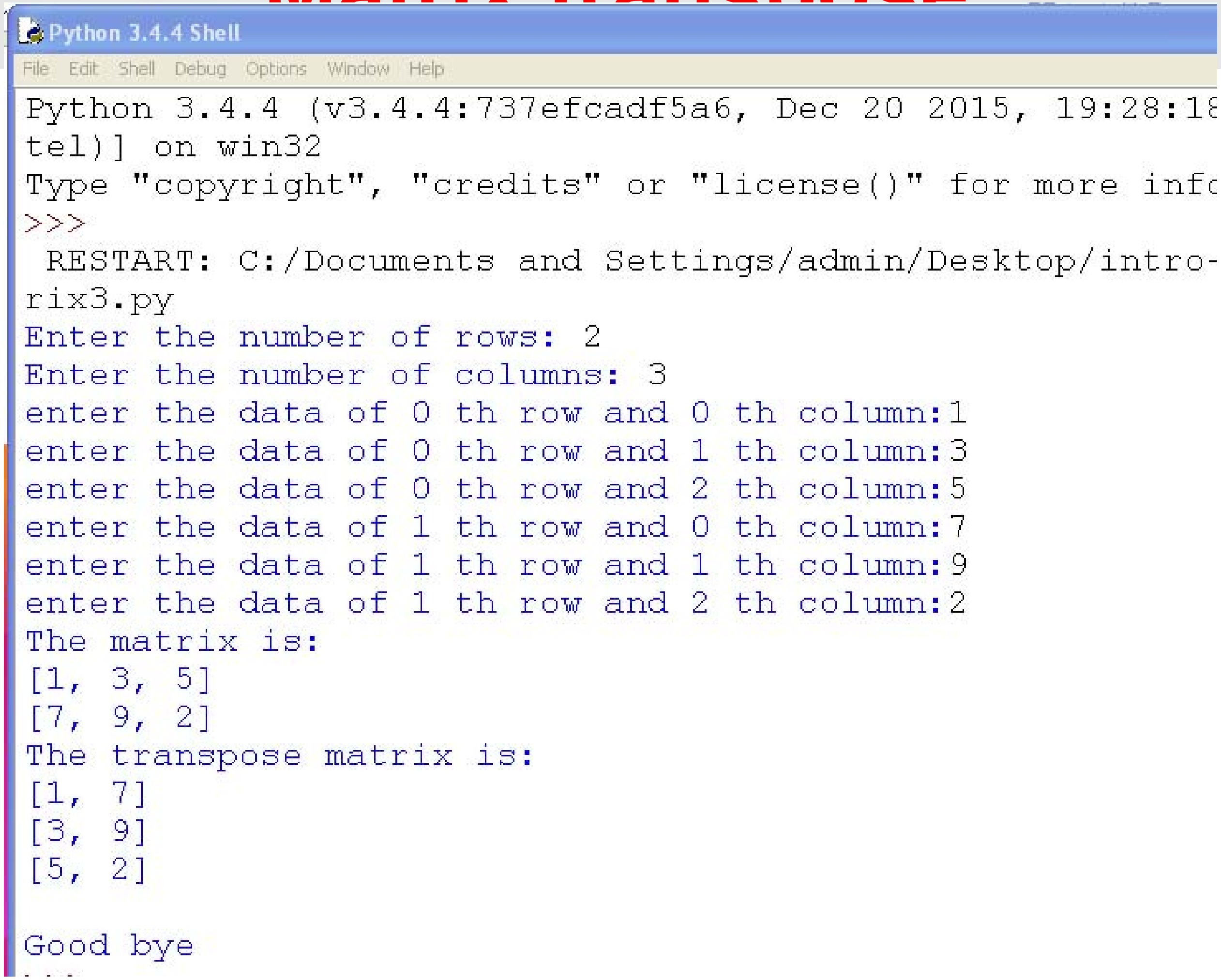
a→ $\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 2 & 0 & 0 \\ 3 & 0 & 0 \end{array}$

b→ $\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

[0] [0]
[1] [0]
[2] [0]
[3] [0]

[0] [1]
[1] [1]
[2] [1]
[3] [1]

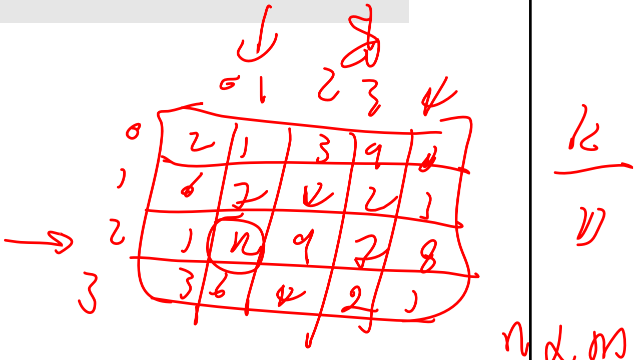
Matrix transpose



```
Python 3.4.4 (v3.4.4:737efcadf5a6, Dec 20 2015, 19:28:18
tel)] on win32
Type "copyright", "credits" or "license()" for more info
>>>
 RESTART: C:/Documents and Settings/admin/Desktop/intro-
rix3.py
Enter the number of rows: 2
Enter the number of columns: 3
enter the data of 0 th row and 0 th column:1
enter the data of 0 th row and 1 th column:3
enter the data of 0 th row and 2 th column:5
enter the data of 1 th row and 0 th column:7
enter the data of 1 th row and 1 th column:9
enter the data of 1 th row and 2 th column:2
The matrix is:
[1, 3, 5]
[7, 9, 2]
The transpose matrix is:
[1, 7]
[3, 9]
[5, 2]

Good bye
>>>
```

Matrix Search

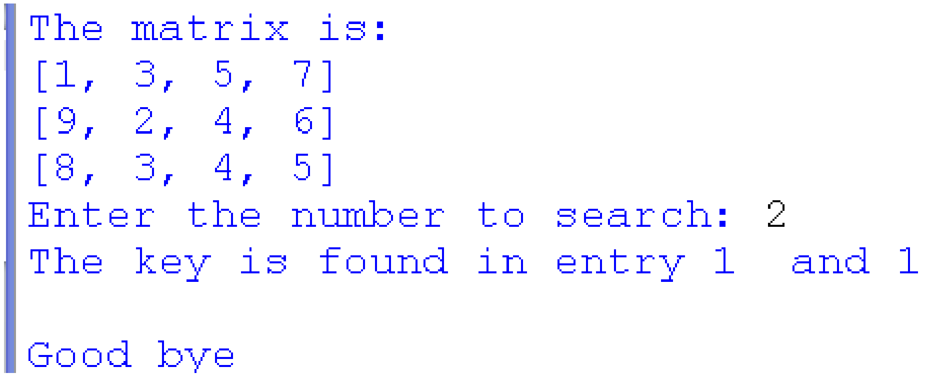
```
        print (a[i])

def matrixSearch(b, k):
    for i in range(len(b)):
        for j in range(len(b[i])):
            if (b[i][j]==k):
                return i,j
    return -1, -1


n=int(input("Enter the number of rows: "))
m=int(input("Enter the number of columns: "))
a=[[0 for j in range(m)] for i in range(n)]
readMatrix(a)
print('The matrix is:')
printMatrix(a)
k=int(input("Enter the number to search: "))
i,j=matrixSearch(a, k)
if (i!=-1):
    print('The key is found in entry %d  and %d' %(i,j))
else:
    print('The key is not found')

print()
print('Good bye')
```



i j

0 0
1
2
3
4
1 0
1
2 3 4

Matrix Search



```
The matrix is:
[1, 3, 5, 7]
[9, 2, 4, 6]
[8, 3, 4, 5]
Enter the number to search: 2
The key is found in entry 1  and 1

Good bye
```

Matrix by array multiplication

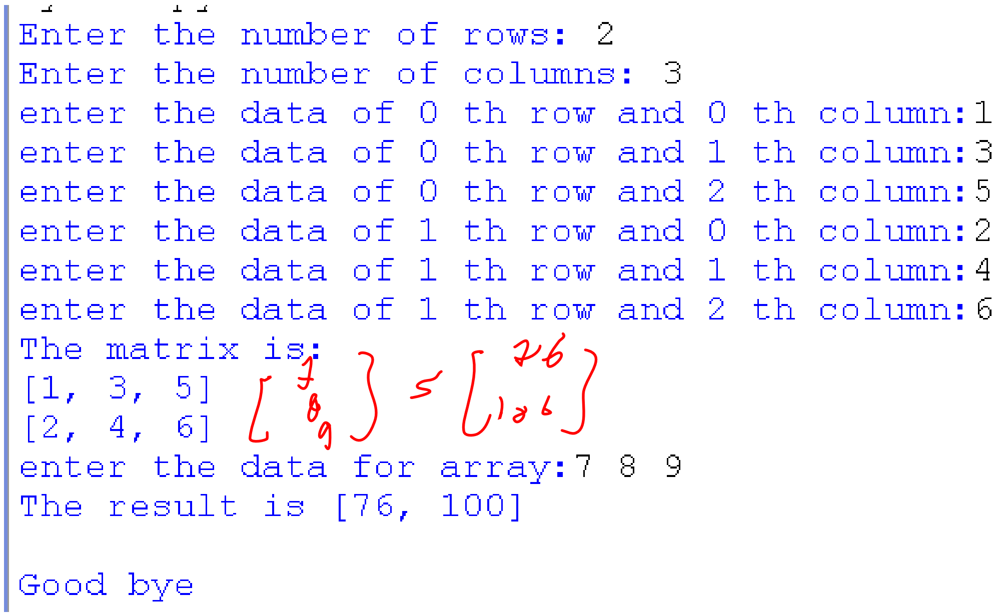
```
def matrixArrayMult(b, x):
    c=[0] * len(b)
    for i in range(len(b)):
        for j in range(len(b[i])):
            c[i]=c[i]+ b[i][j] * x[j]
    return c

n=int(input("Enter the number of rows: "))
m=int(input("Enter the number of columns: "))
b=[[0 for j in range(m)] for i in range(n)]
readMatrix(b)
print('The matrix is:')
printMatrix(b)

x=[]
print("enter the data for array:", end='')
x=list(map(int,input().strip().split(" ")))
c=matrixArrayMult(b, x)
print('The result is', c)

print()
print('Good bye')
```

$$b \quad \begin{array}{c|ccccc} & 0 & 1 & 2 & 3 & 4 \\ \hline 0 & 1 & 2 & 1 & 3 & 1 \\ 1 & 4 & 2 & 1 & 5 & 6 \\ 2 & 3 & 3 & 7 & 8 & 1 \\ 3 & 4 & 1 & 2 & 9 & 2 \end{array} \qquad x \quad \begin{array}{|c|} \hline 2 \\ 4 \\ 3 \\ 6 \\ 1 \\ \hline \end{array} \qquad c \quad \begin{array}{|c|} \hline \cancel{3} \\ 0 \\ 6 \\ 2 \\ 1 \\ \hline \end{array}$$

n x m m x 1 n x 1

i	j	c[i] = c[i] + —
0	0	c[0] = c[0] + b[0][0] x[0]
	↓	c[0] = c[0] + b[0][1] x[1]
	2	c[0] = c[0] + b[0][2] x[2]
↓		

Matrix by array multiplication



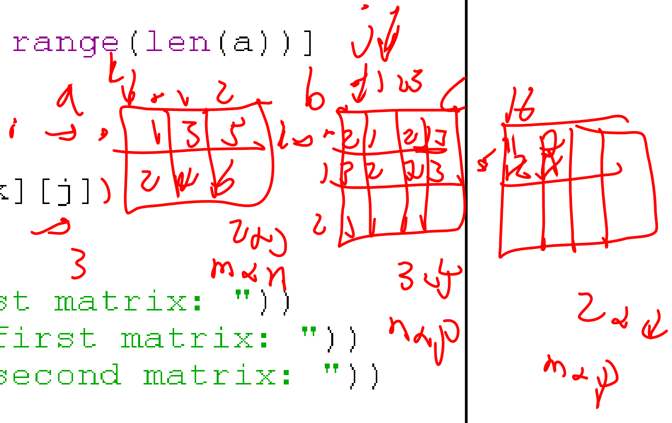
```
Enter the number of rows: 2
Enter the number of columns: 3
enter the data of 0 th row and 0 th column:1
enter the data of 0 th row and 1 th column:3
enter the data of 0 th row and 2 th column:5
enter the data of 1 th row and 0 th column:2
enter the data of 1 th row and 1 th column:4
enter the data of 1 th row and 2 th column:6
The matrix is:
[1, 3, 5]
[2, 4, 6]
enter the data for array:7 8 9
The result is [76, 100]

Good bye
```

$\begin{bmatrix}7\\8\\9\end{bmatrix} = \begin{bmatrix}76\\126\end{bmatrix}$

Matrix by matrix multiplication

```
def matrixMult(a, b):
    c=[[0 for j in range(len(b[0]))] for i in range(len(a))]
    for i in range(len(a)):
        for j in range(len(b[0])):
            for k in range(len(b)):
                c[i][j]=c[i][j]+ a[i][k] * b[k][j])
    return c

n=int(input("Enter the number of rows for first matrix: "))
m=int(input("Enter the number of columns for first matrix: "))
p=int(input("Enter the number of columns for second matrix: "))
a=[[0 for j in range(m)] for i in range(n)]
b=[[0 for j in range(p)] for i in range(m)]
print('first matrix read:')
readMatrix(a)
print('second matrix read:')
readMatrix(b)
print('The first matrix is:')
printMatrix(a)
print('The second matrix is:')
printMatrix(b)
c=matrixMult(a, b)
print('The result is:')
printMatrix(c)

print()
print('Good bye')
```

n3
m2
n
O(m*p*n)



i	j	k
0	0	0
		1
		2
0	1	0
		1
1	2	2

c[0][0] = c[0][0] + a[0][0] * b[0][0]

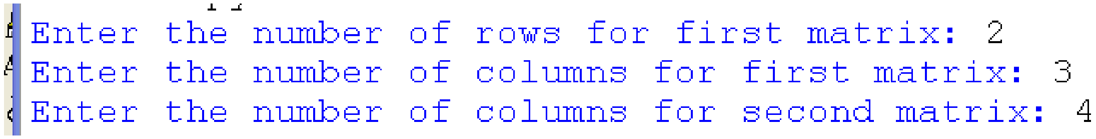
c[0][0] = c[0][0] + a[0][k] * b[1][0]

c[0][0] = c[0][0] + a[0][2] * b[2][0]

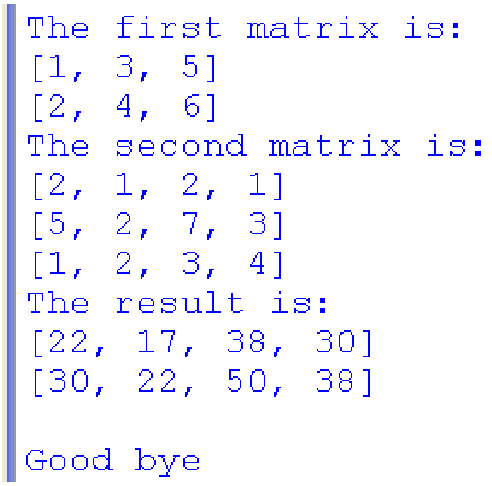
c[0][1] = c[0][1] + a[0][0] * b[0][1]

+ a[0][1] * b[1][1]

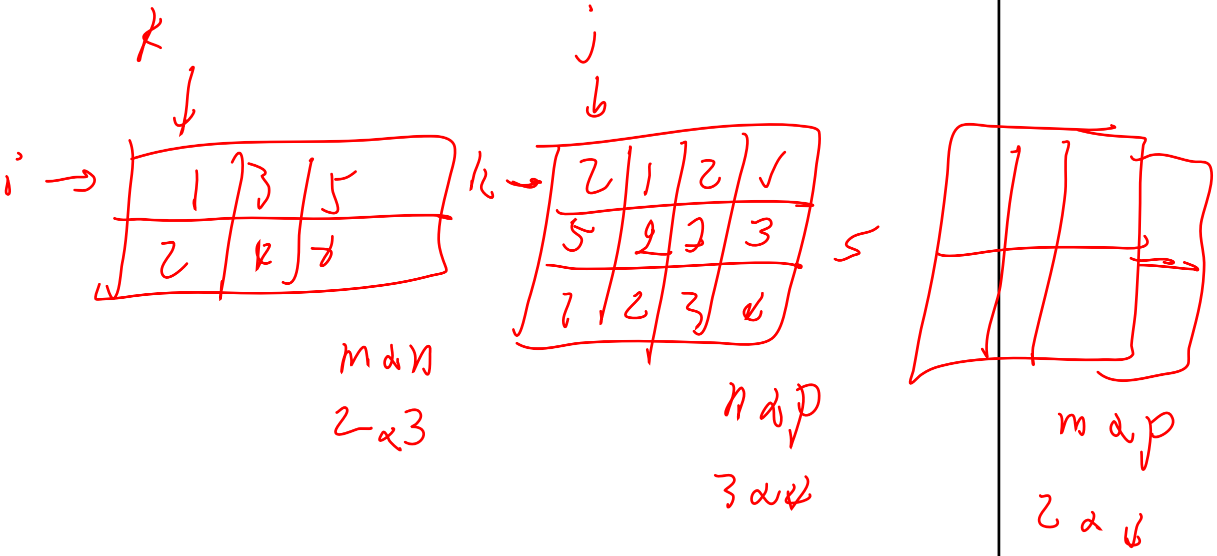
Matrix by matrix multiplication



```
Enter the number of rows for first matrix: 2
Enter the number of columns for first matrix: 3
Enter the number of columns for second matrix: 4
```



```
The first matrix is:
[1, 3, 5]
[2, 4, 6]
The second matrix is:
[2, 1, 2, 1]
[5, 2, 7, 3]
[1, 2, 3, 4]
The result is:
[22, 17, 38, 30]
[30, 22, 50, 38]

Good bye
```



End