Add number on to Table

Write functions that create lists within a list (representing a table) as shown below.

```
def pattern1(nrows, ncols):
      # nrows ≥ 0, ncols ≥ 0


def pattern2(nrows, ncols):
      # nrows ≥ 0, ncols ≥ 0


def pattern3( N ):          # N ≥ 0




def pattern4( N ):          # N ≥ 0




def pattern5( N ):          # N ≥ 0




def pattern6( N ):          # N ≥ 0




exec(input().strip())
```

pattern1(3,7) =

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21

Pattern2(3,7) =

1	4	7	10	13	16	19
2	5	8	11	14	17	20
3	6	9	12	15	18	21

Pattern3(5) =

1	2	3	4	5
0	6	7	8	9
0	0	10	11	12
0	0	0	13	14
0	0	0	0	15

Pattern4(5) =

1	3	6	10	15
0	2	5	9	14
0	0	4	8	13
0	0	0	7	12
0	0	0	0	11

Pattern5(5) =

1	6	10	13	15
0	2	7	11	14
0	0	3	8	12
0	0	0	4	9
0	0	0	0	5

Pattern6(5) =

1	9	10	14	15
0	2	8	11	13
0	0	3	7	12
0	0	0	4	6
0	0	0	0	5

Code template as listed here.

```
def pattern1(nrows, ncols): #nrows >=0, ncols >= 0
def pattern2(nrows, ncols): #nrows >=0, ncols >= 0
def pattern3(N): #N >=0
def pattern4(N): #N >=0
def pattern5(N): #N >=0
def pattern6(N): #N >=0
exec(input().strip())
```

Input

Command in Python language to test a function.

Output

Result of the function execution.

Example

Input (from keyboard)	Output (on screen)
`print(pattern1(3,4))`	`[[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]]`
`print(pattern2(3,4))`	`[[1, 4, 7, 10], [2, 5, 8, 11], [3, 6, 9, 12]]`
`print(pattern3(4))`	`[[1, 2, 3, 4], [0, 5, 6, 7], [0, 0, 8, 9], [0, 0, 0, 10]]`
`print(pattern4(4))`	`[[1, 3, 6, 10], [0, 2, 5, 9], [0, 0, 4, 8], [0, 0, 0, 7]]`
`print(pattern5(4))`	`[[1, 5, 8, 10], [0, 2, 6, 9], [0, 0, 3, 7], [0, 0, 0, 4]]`
`print(pattern6(4))`	`[[1, 7, 8, 10], [0, 2, 6, 9], [0, 0, 3, 5], [0, 0, 0, 4]]`