Slicing & element-wise operation

Your task is to implement these function (detail is in the comment)

```
import numpy as np

def sum_2_rows( M ):
    #  Return summation in each column of 2 adjacent rows.
    #  Ex M = [[ 0,  1,  2,  3],   is   [[ 4,  6,  8, 10],
    #          [ 4,  5,  6,  7],         [20, 22, 24, 26]]
    #          [ 8,  9, 10, 11],
    #          [12, 13, 14, 15]]


def sum_left_right( M ):
    #  Return summation of left half and right half of M
    #  Ex M = [[ 0,  1,  2,  3],   is   [[ 2,  4],
    #          [ 4,  5,  6,  7],         [10, 12],
    #          [ 8,  9, 10, 11],         [18, 20],
    #          [12, 13, 14, 15]]         [26, 28]]


def sum_upper_lower( M ):
    #  Return summation of upper half and lower half of M
    #  Ex M = [[ 0,  1,  2,  3],   is   [[ 8, 10, 12, 14],
    #          [ 4,  5,  6,  7],         [16, 18, 20, 22]]
    #          [ 8,  9, 10, 11],
    #          [12, 13, 14, 15]]
```

```
def sum_4_quadrants( M ):
    #  Return summation in the same position from 4 quadrants
    #  Ex M = [[ 0,  1,  2,  3],   is   [[20, 24],
    #          [ 4,  5,  6,  7],         [36, 40]]
    #          [ 8,  9, 10, 11],
    #          [12, 13, 14, 15]]


def sum_4_cells( M ):
    #  Return summation of 4 adjacent numbers according to the pattern below.
    #  Ex M = [[ 0,  1,  2,  3],   is   [[10, 18],
    #          [ 4,  5,  6,  7],         [42, 50]]
    #          [ 8,  9, 10, 11],
    #          [12, 13, 14, 15]]

def count_leap_years( years ):
    #  Years is array which contains Buddhist years
    #  Return the number of leap years (years which have 366 days) in years


exec(input().strip())      # This command is necessary to grade your answer
```

Input

Python commands which are used to test the functions.

Output

Result from the program.

Example

Input (from keyboard)	Output (on screen)

`print(sum_2_rows(np.arange(36).reshape(6,6)))`	`[[ 6 8 10 12 14 16]`
	`[30 32 34 36 38 40]`
	`[54 56 58 60 62 64]]`

`print(sum_left_right(np.arange(36).reshape(6,6)))`	`[[ 3 5 7]`
	`[15 17 19]`
	`[27 29 31]`
	`[39 41 43]`
	`[51 53 55]`
	`[63 65 67]]`

`print(sum_upper_lower(np.arange(36).reshape(6,6)))`	`[[18 20 22 24 26 28]`
	`[30 32 34 36 38 40]`
	`[42 44 46 48 50 52]]`

`print(sum_4_quadrants(np.arange(36).reshape(6,6)))`	`[[42 46 50]`
	`[66 70 74]`
	`[90 94 98]]`

`print(sum_4_cells(np.arange(36).reshape(6,6)))`	`[[ 14 22 30]`
	`[ 62 70 78]`
	`[110 118 126]]`

`print(count_leap_years(np.array([2543,2559,2560])))`	`2`