

Input Of Two Dimensional Lists

Row Wise Sum

Given a 2D integer array of size $M \times N$, find and print the sum of i th row elements separated by space.

Input Format :

Line 1 : Two integers M and N (separated by space)

Line 2 : Matrix elements of each row (separated by space)

Output Format :

Sum of every i th row elements (separated by space)

Constraints :

$0 \leq N \leq 10^4$

$0 \leq M \leq 10^4$

Sample Input 1:

4 2

1 2

3 4
5 6
7 8

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In []:

Handwritten notes:

- `[[], [], []]`
- `[int(j) for j in input().split()]`
- `n, m =`
- Diagram of a 3x3 grid with values 1, 3, 4 in the top row and 2, 2, 4 in the bottom row.
- Diagram of a 3x3 grid with values 3, 4 in the top row, 1, 2, 3, 4 in the middle row, 5, 6, 7, 8 in the bottom row, and 9, 10, 11, 12 in the bottom row.

localhost:8888/notebook/python2dlist/input2dlist.py Jupyter input2dlist Last Checkpoint: 8 minutes ago (autosaved) Logout Python 3

Handwritten notes:

- `[[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]]`
- `[int(j) for j in input().split()]`
- `for i in range(n)`
- Diagram of a 3x3 grid with values 1, 2, 3, 4 in the top row, 5, 6, 7, 8 in the middle row, and 9, 10, 11, 12 in the bottom row.

```
In [27]: s = input().split()
         n,m = int(s[0]), int(s[1])
```

3 4

```
In [28]: # last list for run loops
         l_2_d = [ "hello" for i in range(n)]
         print(l_2_d)
```

['hello', 'hello', 'hello']

last list for run loops, first loop for print lists

```
In [29]: l_2_d = [ [] for i in range(n)]  
          print(l_2_d)
```

```
[[], [], []]
```

By default it is string

```
In [9]: l_2_d = [ [ele for ele in input().split()] for i in range(n)]  
          print(l_2_d)
```

```
1 2 3 4  
5 6 7 8  
9 10 11 12  
[['1', '2', '3', '4'], ['5', '6', '7', '8'], ['9', '10', '11', '12']]
```

```
In [10]: l_2_d = [ [int(ele) for ele in input().split()] for i in range(n)]  
          l_2_d
```

```
1 2 3 4  
5 6 7 8  
9 10 11 12
```

```
Out[10]: [[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]]
```

```
In [11]: print(l_2_d)
```

```
[[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]]
```

```
In [13]: j = [print(i) for i in l_2_d]
```

```
[1, 2, 3, 4]  
[5, 6, 7, 8]  
[9, 10, 11, 12]
```

```
In [14]: j
```

```
Out[14]: [None, None, None]
```

```
In [15]: j = [i for i in l_2_d]  
j
```

```
Out[15]: [[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]]
```

```
In [18]: j = [print(*i) for i in l_2_d]  
j
```

```
1 2 3 4  
5 6 7 8  
9 10 11 12
```

```
Out[18]: [None, None, None]
```

Input of jagged list: only take row input, no use of column

```
In [20]: n = int(input())  
         l_2_d = [[int(ele) for ele in input().split()] for i in range(n)]
```

```
5  
1 2  
1 2 4 4  
1 3 4  
1  
1 2 3 4 5
```

```
In [21]: l_2_d
```

```
Out[21]: [[1, 2], [1, 2, 4, 4], [1, 3, 4], [1], [1, 2, 3, 4, 5]]
```

```
In [22]: print(*l_2_d)
```

```
[1, 2] [1, 2, 4, 4] [1, 3, 4] [1] [1, 2, 3, 4, 5]
```



```
In [23]: j = [print(i) for i in l_2_d]
```

```
[1, 2]  
[1, 2, 4, 4]  
[1, 3, 4]  
[1]  
[1, 2, 3, 4, 5]
```

```
In [24]: j = [print(*i) for i in l_2_d]
```

```
1 2  
1 2 4 4  
1 3 4  
1  
1 2 3 4 5
```

Another format taking inputs of 2-d array

Row Wise Sum

Given a 2D integer array of size M*N, find and print the sum of ith row elements separated by space.

Input Format :

Line 1 : Two integers M and N (separated by space)

Line 2 : Matrix elements of each row (separated by space)

Output Format :

Sum of every ith row elements (separated by space)

Constraints :

$0 \leq N \leq 10^4$

$0 \leq M \leq 10^4$

Sample Input 1:

4 2

1 2 3 4 5 6 7 8

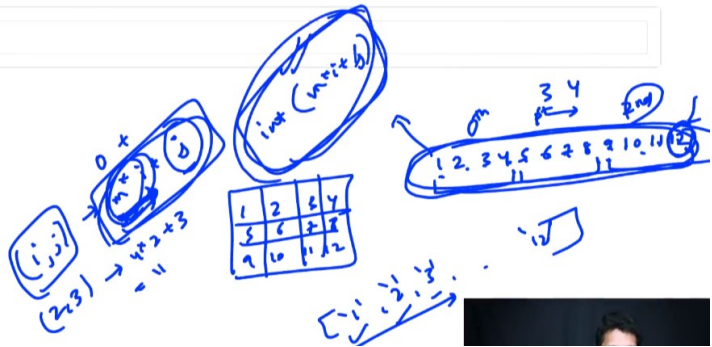
lists of lists : m*i + j

run outer loop n times

run inner loop column times

int(2_d_i[m*i+j]): for which element you want from array input
string

In []:



```
In [32]: s = input().split()
n,m = int(s[0]), int(s[1])

# take all array elements in one line.
arr_i = input().split()

arr = [ [ int(arr_i[m*i+j]) for j in range(m)] for i in range(n)]

print(arr)

3 4
1 2 3 4 5 6 7 8 9 10 11 12
[[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]]
```

```
In [33]: arr
```

```
Out[33]: [[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]]
```

```
In [38]: [print(i) for i in arr]
```

```
[1, 2, 3, 4]  
[5, 6, 7, 8]  
[9, 10, 11, 12]
```

```
Out[38]: [None, None, None]
```

Another input form

```
row input array  
4 2 1 2 3 4 5 6 7 8
```

```
In [39]: arr_str = input().split()
          n,m = int(arr_str[0]), int(arr_str[1])
          arr_in = arr_str[2:]
          # take array input
          arr = [ [ int(arr_in[m*i + j]) for j in range(m)] for i in range(n) ]

          arr
```

```
3 4 1 2 3 4 5 6 7 8 9 10 11 12
```

```
Out[39]: [[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]]
```

```
In [40]: print(arr)
```

```
[[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]]
```

```
In [41]: print(n, m )
```

```
3 4
```

Printing 2-D Array: Iterate on 2-D list

```
In [42]: arr
```

```
Out[42]: [[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]]
```

```
In [43]: n
```

```
Out[43]: 3
```

```
In [44]: m
```

```
Out[44]: 4
```

```
In [46]: # printing 2-d array  
         for i in arr:  
             print(i)
```

```
[1, 2, 3, 4]
```

```
[5, 6, 7, 8]
```

```
[9, 10, 11, 12]
```

```
In [57]: # printing 2-d array
        for i in range(n):
            for j in range(m):
                # each row in one line , break after one row
                print(arr[i][j],end=" ")
            print()
```

```
1 2 3 4
5 6 7 8
9 10 11 12
```

Printing Jagged List

```
In [58]: jag = [[1, 2], [1, 2, 4, 4], [1, 3, 4], [1], [1, 2, 3, 4, 5]]
```

```
In [66]: jag
```

```
Out[66]: [[1, 2], [1, 2, 4, 4], [1, 3, 4], [1], [1, 2, 3, 4, 5]]
```



```
In [68]: n = 5
```

```
In [69]: # printing jagged list  
# for row  
for i in range(n):  
    #for column  
    for j in j[i]:  
        print(j, end=" ")  
    print()
```

```
1 2  
1 2 4 4  
1 3 4  
1  
1 2 3 4 5
```

```
In [70]: # another way
         # iterate over main list
         for i in jag:
             # iterate over inner lists
             for j in i:
                 print(j, end=" ")

             print()
```

```
1 2
1 2 4 4
1 3 4
1
1 2 3 4 5
```

Using Join()

```
In [71]: "ab".join("-")
```

```
Out[71]: '-'
```

Not join after last element

```
In [74]: "-".join("abc")
```

```
Out[74]: 'a-b-c'
```

```
In [78]: "ab".join([1,2,3])
```

```
-----  
-----  
TypeError                                Traceback (most recent c  
all last)  
<ipython-input-78-63a96a800c83> in <module>  
----> 1 "ab".join([1,2,3])  
  
TypeError: sequence item 0: expected str instance, int found
```

```
In [79]: "ab".join("1234")
```

```
Out[79]: '1ab2ab3ab4'
```

```
In [80]: "ab".join("abcd")
```

```
Out[80]: 'aabbabcabd'
```

String cant concate with list

```
In [83]: "ab".join([1,2,3])
```

```
-----  
-----  
TypeError                                 Traceback (most recent c  
all last)  
<ipython-input-83-63a96a800c83> in <module>  
----> 1 "ab".join([1,2,3])  
  
TypeError: sequence item 0: expected str instance, int found
```

```
In [84]: "ab".join(['1','2','3'])
```

```
Out[84]: '1ab2ab3'
```

```
In [85]: jag
```

```
Out[85]: [[1, 2], [1, 2, 4, 4], [1, 3, 4], [1], [1, 2, 3, 4, 5]]
```

```
In [88]: str(jag)
```

```
Out[88]: '[[1, 2], [1, 2, 4, 4], [1, 3, 4], [1], [1, 2, 3, 4, 5]]'
```

```
In [86]: n
```

```
Out[86]: 5
```

Join operation not working on last element of iterable.

```
In [94]: for i in jag:
          out = " ".join([str(x) for x in i])
          print(out)
```

```
1 2
1 2 4 4
1 3 4
1
1 2 3 4 5
```

```
In [ ]:
```