Colab: https://colab.research.google.com/drive/1pTJxkzMoczqgwXabPGSeFvUTyzcaktkS?usp=sharing

```
import numpy as np
```

fit.txt

```
#date   step_count      mood    calories_burned hours_of_sleep  active
06-10-2017      5464    Neutral 181     5       Inactive
07-10-2017      6041    Sad     197     8       Inactive
08-10-2017      25      Sad     0       5       Inactive
09-10-2017      5461    Sad     174     4       Inactive
10-10-2017      6915    Neutral 223     5       Active
11-10-2017      4545    Sad     149     6       Inactive
12-10-2017      4340    Sad     140     6       Inactive
13-10-2017      1230    Sad     38      7       Inactive
14-10-2017      61      Sad     1       5       Inactive
15-10-2017      1258    Sad     40      6       Inactive
16-10-2017      3148    Sad     101     8       Inactive
17-10-2017      4687    Sad     152     5       Inactive
18-10-2017      4732    Happy   150     6       Active
19-10-2017      3519    Sad     113     7       Inactive
20-10-2017      1580    Sad     49      5       Inactive
21-10-2017      2822    Sad     86      6       Inactive
22-10-2017      181     Sad     6       8       Inactive
23-10-2017      3158    Neutral 99      5       Inactive
24-10-2017      4383    Neutral 143     4       Inactive
25-10-2017      3881    Neutral 125     5       Inactive
26-10-2017      4037    Neutral 129     6       Inactive
```

```
m1 = np.array([[1,2,3],
               [4,5,6]])
m1
```

```
array([[1, 2, 3],
       [4, 5, 6]])
```

```
m1.ndim
```

```
2
```

```
m1.shape
```

```
(2, 3)
```

```
len(m1)
```

```
2
```

```
m2 = np.arange(1, 13)
```

```
m2.shape
```

```
(12,)
```

```
m2.reshape(3, 4)
```

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

```
m2.reshape(4, 3)
```

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

```
m2.reshape(1, 12)
```

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

```
m2.reshape(12, 1)
```

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

```
m = np.array([[ 1,  2,  3],
       [ 4,  5,  6],
       [ 7,  8,  9],
       [10, 11, 12]])
m
```

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

```
m.T
```

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

```
a = np.arange(3)
a
```

```
array([0, 1, 2])
```

```
a.T
```

```
array([0, 1, 2])
```

```
a = np.arange(3).reshape(1, 3)
a
```

```
array([[0, 1, 2]])
```

```
a.T
```

```
array([[0],
       [1],
       [2]])
```

```
A = np.arange(12).reshape(3, 4)
A
```

```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11]])
```

```
A.flatten()
```

```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11])
```

```
m1 = np.arange(1,10).reshape((3,3))
m1
```

```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

```
m1[1][2] # 2-step process
```

```
6
```

```
m1[1, 2] # 1-step process
```

```
6
```

```
m1 = np.array([100,200,300,400,500,600])
m1[[2,3,4,1,2,2]]
```

```
array([300, 400, 500, 200, 300, 300])
```

```
m1
```

```
array([100, 200, 300, 400, 500, 600])
```

```
m1 = np.arange(1,10).reshape((3,3))
m1
```

```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

```
m1[[0, 1, 2], [0, 1, 2]]
```

```
array([1, 5, 9])
```

```
m1 = np.arange(12).reshape(3,4)
m1
```

```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11]])
```

```
m1[:2]
```

```
array([[0, 1, 2, 3],
       [4, 5, 6, 7]])
```

```
m1[:2, :]
```

```
array([[0, 1, 2, 3],
       [4, 5, 6, 7]])
```

```
m1 = np.arange(12).reshape(3,4)
m1
```

```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11]])
```

```
# [[5, 6],
#  [9, 10]]
m1[1:3, 1:3]
```

```
array([[ 5,  6],
       [ 9, 10]])
```

```
# [[1, 3],
#  [5, 7],
#  [9,11]]
m1[:, [1, 3]]
```

```
array([[ 1,  3],
       [ 5,  7],
       [ 9, 11]])
```

```
m1[:, 1::2]
```

```
array([[ 1,  3],
       [ 5,  7],
       [ 9, 11]])
```

```
# Fancy Indexing
```

```
m1 = np.arange(12).reshape(3, 4)
m1
```

```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11]])
```

```
m1 < 6
```

```
array([[ True,  True,  True,  True],
       [ True,  True, False, False],
       [False, False, False, False]])
```

```
m1[m1 < 6]
```

```
array([0, 1, 2, 3, 4, 5])
```

```
# universal functions
```

```
a = np.arange(12).reshape(3, 4)
a
```

```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11]])
```

```
np.sum(a)
```

```
66
```

```
np.sum(a, axis=1)
```

```
array([ 6, 22, 38])
```

```
np.sum(a, axis=0)
```

```
array([12, 15, 18, 21])
```

```
np.mean(a, axis=1)
```

```
array([1.5, 5.5, 9.5])
```

```
a = np.arange(12).reshape(3, 4)
a
```

```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11]])
```

```
np.min(a, axis=0)
```

```
array([0, 1, 2, 3])
```

```
# Logical Operations
```

```
a = np.array([1,2,3,4])
b = np.array([4,3,2,1])
a, b
```

```
(array([1, 2, 3, 4]), array([4, 3, 2, 1]))
```

```
a < b
```

```
array([ True,  True, False, False])
```

```
np.any(a < b)
```

```
True
```

```
np.all(a < b)
```

```
False
```

```
a = np.array([1, 2, 3, 2])
b = np.array([2, 2, 3, 2])
c = np.array([6, 4, 4, 5])
np.all(((a <= b) & (b <= c)))
```

```
True
```

```
a = np.array([1,2,3,4])
b = np.array([4,3,2,1])
np.sum(a < b)
```

```
2
```

```
arr = np.array([-3,4,27,34,-2, 0, -45,-11,4, 0 ])
arr >= 0 # positive ---> 1 or negative ----> -1
```

```
array([False,  True,  True,  True, False,  True, False, False,  True,
        True])
```

```
# arr[arr >= 0] = 1
# arr[arr < 0]  = -1
np.where(arr >=0, 1, -1)
```

```
array([-1,  1,  1,  1, -1,  1, -1, -1,  1,  1])
```

```
a = np.array([2,30,41,7,17,52])
a
```

```
array([ 2, 30, 41,  7, 17, 52])
```

```
np.sort(a)
```

```
array([ 2,  7, 17, 30, 41, 52])
```

```
a = np.arange(9,0,-1).reshape(3,3)
a
```

```
array([[9, 8, 7],
       [6, 5, 4],
       [3, 2, 1]])
```

```
np.sort(a, axis=0)
```

```
array([[3, 2, 1],
       [6, 5, 4],
       [9, 8, 7]])
```

```
np.sort(a, axis=1)
```

```
array([[7, 8, 9],
       [4, 5, 6],
       [1, 2, 3]])
```

```
a = np.array([2,30,41,7,17,52])
a
```

```
array([ 2, 30, 41,  7, 17, 52])
```

```
np.argmax(a)
```

```
5
```

```
np.argsort(a)
```

```
array([0, 3, 4, 1, 2, 5])
```

```
# solve the fitbit use-case
```

✓ 0s completed at 23:06