Print the product of NESTLE company with quantity

Reading / Writing Data and Handling the missing Data# New Section

```
string_data = pd.Series(["Amazon", np.nan, None, "Flipkart"])
string_data
```

```
0      Amazon
1         NaN
2        None
3    Flipkart
dtype: object
```

```
string_data.isna()
```

```
0    False
1     True
2     True
3    False
dtype: bool
```

```
data = pd.Series([1, np.nan, 7.5, np.nan, 15])
data.dropna()
```

```
0     1.0
2     7.5
4    15.0
dtype: float64
```

```
data[data.notna()]
```

```
0     1.0
2     7.5
4    15.0
dtype: float64
```

```
data.dropna(how="all")
```

```
0     1.0
2     7.5
4    15.0
dtype: float64
```

```
import numpy as np

data = pd.DataFrame(np.arange(6).reshape((2, 3)),
index=pd.Index(['Snacks', 'Drinks'], name='products'),
```

```
columns=pd.Index(['23', '55', '12'],name='Quantity'))
data
```

Quantity	23	55	12
products			
Snacks	0	1	2
Drinks	3	4	5

```
result = data.stack()
print(result)
result.unstack()
```

```
products  Quantity
Snacks    23          0
          55          1
          12          2
Drinks    23          3
          55          4
          12          5
dtype: int64
```

Quantity	23	55	12
products			
Snacks	0	1	2
Drinks	3	4	5

```
df = pd.DataFrame({'Chocolates': ['Dark', 'White', 'Peanuts'],
'Milkybar': [1, 2, 3],
'Kitkat': [4, 5, 6],
'Fuse': [7, 8, 9]})
df
```

	Chocolates	Milkybar	Kitkat	Fuse
0	Dark	1	4	7
1	White	2	5	8
2	Peanuts	3	6	9

```
df.iloc[4:, 2] = np.nan
```

```
data = pd.DataFrame({"k1": ["one", "two"] *3+ ["two"],
                     "k2": [1, 1, 2, 3, 3, 4, 4]})
```

```
data
```

	k1	k2
0	one	1
1	two	1
2	one	2
3	two	3
4	one	3
5	two	4
6	two	4

```
data.duplicated()
```

```
0    False
1    False
2    False
3    False
4    False
5    False
6     True
dtype: bool
```

```
data.drop_duplicates()
```

	k1	k2
0	one	1
1	two	1
2	one	2
3	two	3
4	one	3
5	two	4

```
data["v1"] =range(7)
data
```

	k1	k2	v1
0	one	1	0
1	two	1	1
2	one	2	2
3	two	3	3

```
import pandas as pd


data = {'Name': ['Nesto', 'Grans Fresh', 'LULU','Hypercity', 'Mukkam Mall', 'KK MALL', 'Gokul
                'Date': [17, 20, 18, 22, 18, 15, 17],
                    'Deliverd': ['y', 'n', 'y', 'n', 'n', 'n', 'y'],
                        'Quantity': [90, 76, 'NaN', 74, 65, 'NaN', 71]}
df = pd.DataFrame(data)
df
```

	Name	Date	Deliverd	Quantity
0	Nesto	17	y	90
1	Grans Fresh	20	n	76
2	LULU	18	y	NaN
3	Hypercity	22	n	74
4	Mukkam Mall	18	n	65
5	KK MALL	15	n	NaN
6	Gokulam Mall	17	y	71

```
c = avg = 0
for ele in df['Quantity']:
    if str(ele).isnumeric():
        c += 1
        avg += ele
avg /= c


df = df.replace(to_replace="NaN",value=avg)
df
```

	Name	Date	Deliverd	Quantity
0	Nesto	17	y	90.0
1	Grans Fresh	20	n	76.0
2	LULU	18	y	75.2

```
data = pd.Series([1., -999., 2., -999., -1000., 3.])
print(data)
data.replace(-999, np.nan)
```

```
0       1.0
1    -999.0
2       2.0
3    -999.0
4   -1000.0
5       3.0
dtype: float64
0       1.0
1       NaN
2       2.0
3       NaN
4   -1000.0
5       3.0
dtype: float64
```

```
ages = [20, 22, 25, 27, 21, 23, 37, 31, 61, 45, 41, 32]
bins = [18, 25, 35, 60, 100]
cats = pd.cut(ages, bins)
cats
```

```
[(18, 25], (18, 25], (18, 25], (25, 35], (18, 25], ..., (25, 35], (60, 100], (35, 60],
(35, 60], (25, 35]]
Length: 12
Categories (4, interval[int64, right]): [(18, 25] < (25, 35] < (35, 60] < (60, 100]]
```

```
print(cats.codes)
cats.categories
```

```
[0 0 0 1 0 0 2 1 3 2 2 1]
IntervalIndex([(18, 25], (25, 35], (35, 60], (60, 100]], dtype='interval[int64,
right]')
```

```
df = pd.DataFrame(np.arange(5 * 4).reshape((5, 4)))
sampler = np.random.permutation(5)
sampler
```

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

```
print(df)
```

```
df.take(sampler)
```

```
    0   1   2   3
0   0   1   2   3
1   4   5   6   7
2   8   9  10  11
3  12  13  14  15
4  16  17  18  19
```

	0	1	2	3
3	12	13	14	15
4	16	17	18	19
1	4	5	6	7
2	8	9	10	11
0	0	1	2	3

```
df = pd.DataFrame({'key': ['b', 'b', 'a', 'c', 'a', 'b'],
'data1': range(6)})
pd.get_dummies(df['key'])
```

	a	b	c
0	0	1	0
1	0	1	0
2	1	0	0
3	0	0	1
4	1	0	0
5	0	1	0

Colab paid products - Cancel contracts here