

В объектах `srs1` и `srs2` приведены данные по компании ПАО "ГМК "Норильский никель" за 2020-2022 годы. В серии `srs1` представлена информация по выручке от реализации металлов в млн рублей, а в `srs2` - информация по выручке от прочей реализации в млн рублей.

```
2022    1130546
2021    1261805
2020    1076125
Name: Выручка от реализации металлов, в млн рублей, dtype: int64

2022     53931
2021     55143
2020     40844
Name: Выручка от прочей реализации, в млн рублей, dtype: int64
```

Сложите эти две серии и получите итоговую выручку по каждому году. Выведите полученную серию на печать.

Sample Input:

Sample Output:

```
2022    1184477
2021    1316948
2020    1116969
dtype: int64
```

```
1 import pandas as pd
2
3 d1 = {'2022': 1130546, '2021': 1261805, '2020': 1076125}
4 srs1 = pd.Series(d1, name='Выручка от реализации металлов, в млн рублей')
5 d2 = {'2022': 53931, '2021': 55143, '2020': 40844}
6 srs2 = pd.Series(d2, name='Выручка от прочей реализации, в млн рублей')
7 # Продолжите написание кода здесь
```

В объектах `srs1` , `srs2` и `srs3` приведены данные по компании ПАО "ГМК "Норильский никель" за 2020-2022 годы. В серии `srs1` представлена информация по выручке в млн рублей, в `srs2` - информация по себестоимости реализованных металлов в млн рублей, в `srs3` - информация по себестоимости прочей реализации в млн рублей.

```
2022    1184477
2021    1316948
2020    1116969
Name: Выручка, в млн рублей, dtype: int64
```

```
2022     426329
2021     372728
2020     324308
Name: Себестоимость реализованных металлов, в млн рублей, dtype: int64
```

```
2022      56137
2021      54862
2020      40482
Name: Себестоимость прочей реализации, в млн рублей, dtype: int64
```

Посчитайте валовую прибыль, которая равна: выручка минус себестоимость. Назовите полученный объект `Series` - 'Валовая прибыль, в млн рублей' и выведите его на печать.

Sample Input:

Sample Output:

```
2022    702011
2021    889358
2020    752179
Name: Валовая прибыль, в млн рублей, dtype: int64
```

```
1 import pandas as pd
2
3 d1 = {'2022': 1184477, '2021': 1316948, '2020': 1116969}
4 srs1 = pd.Series(d1, name='Выручка, в млн рублей')
5 d2 = {'2022': 426329, '2021': 372728, '2020': 324308}
6 srs2 = pd.Series(d2, name='Себестоимость реализованных металлов, в млн рублей')
7 d3 = {'2022': 56137, '2021': 54862, '2020': 40482}
8 srs3 = pd.Series(d3, name='Себестоимость прочей реализации, в млн рублей')
9 # Продолжите написание кода здесь
```

В объектах `srs1` и `srs2` приведены данные по нескольким российским металлургическим компаниям. В серии `srs1` представлена информация о количестве акций компании, в `srs2` - информация по рыночной цене акций по итогам торгов на фондовом рынке ПАО Московская Биржа на конец 3-го квартала 2023 года.

```
ПАО "Ашинский метзавод"      498454822
ПАО "Северсталь"              837718660
ПАО "ЧМК"                     3161965
ПАО "ГМК "Норильский никель"  152863397
ПАО "ММК"                     11174330000
ПАО "НЛМК"                    5993227240
ПАО "ТМК"                     1033135366
Name: Объем выпуска, штук, dtype: int64

ПАО "Ашинский метзавод"      103.28
ПАО "Северсталь"              1369.20
ПАО "ЧМК"                     11195.00
ПАО "ГМК "Норильский никель"  16188.00
ПАО "ММК"                     52.26
ПАО "НЛМК"                    209.56
ПАО "ТМК"                     222.14
Name: Рыночная цена на 29.09.2023 г., руб., dtype: float64
```

- Рассчитайте рыночную капитализацию компаний на конец 3-го квартала 2023 года.
- Итоговый объект `Series` должен иметь данные целочисленного типа.
- Назовите итоговую серию: `'Рыночная капитализация на конец 3 квартала 2023 года'`.
- Выведите результат на печать.

Sample Input:

Sample Output:

```
ПАО "Ашинский метзавод"      51480414016
ПАО "Северсталь"              1147004389272
ПАО "ЧМК"                     35398198175
ПАО "ГМК "Норильский никель"  2474552670636
ПАО "ММК"                     583970485800
ПАО "НЛМК"                    1255940700414
ПАО "ТМК"                     229500690203
Name: Рыночная капитализация на конец 3 квартала 2023 года, dtype: int64
```

```
1 import pandas as pd
2
3 d1 = {'ПАО "Ашинский метзавод"': 498454822, 'ПАО "Северсталь"': 837718660,
4       'ПАО "ЧМК"': 3161965, 'ПАО "ГМК "Норильский никель"': 152863397,
5       'ПАО "ММК"': 11174330000, 'ПАО "НЛМК"': 5993227240, 'ПАО "ТМК"': 1033135366}
6 d2 = {'ПАО "Ашинский метзавод"': 103.28, 'ПАО "Северсталь"': 1369.2,
7       'ПАО "ЧМК"': 11195, 'ПАО "ГМК "Норильский никель"': 16188,
8       'ПАО "ММК"': 52.26, 'ПАО "НЛМК"': 209.56, 'ПАО "ТМК"': 222.14}
9 srs1 = pd.Series(d1, name='Объем выпуска, шт.')
10 srs2 = pd.Series(d2, name='Рыночная цена на 29.09.2023 г., руб.')
11 # Продолжите написание кода здесь
```

P/E = Капитализация / Чистая прибыль

P/E является самым популярным мультипликатором, который сравнивает стоимость компании с главным итогом деятельности - чистой прибылью. В классическом варианте для сравнения берутся капитализация за последний торговый день года и чистая прибыль за этот год.

По сути, отношений цены к чистой прибыли показывает сумму в рублях, которую инвестор может инвестировать в компанию, чтобы получить один рубль прибыли этой компании.

*При сравнении двух компаний **P/E** показывает, насколько рубль чистой прибыли одной компании оценивается инвесторами выше, чем рубль чистой прибыли другой. При прочих равных условиях интереснее та компания, у которой значение **P/E** меньше, а значит есть вероятность того, что она недооценена и ее акции могут расти быстрее.*

В объектах `srs1` и `srs2` приведены данные по трём российским энергетическим компаниям за 2022 год. В серии `srs1` представлена информация по чистой прибыли, в `srs2` - информация по капитализации на последний торговый день 2022 года.

```
ПАО "Россети"      54210000000
ПАО "РусГидро"     19325000000
ПАО "Юнипро"       21267722000
Name: Чистая прибыль за 2022 год, руб., dtype: int64
```

```
ПАО "Россети"      1.108449e+11
ПАО "РусГидро"     3.332006e+11
ПАО "Юнипро"       8.486356e+10
Name: Капитализация на конец 2022 года, в млн рублей, dtype: float64
```

Создайте новый объект `Series`, который будет содержать данные по P/E, назовите его: `P/E 2022 год`, округлите значения до **одного** знака после запятой и выведите полученную серию на печать.

Sample Input:

Sample Output:

```
ПАО "Россети"      2.0
ПАО "РусГидро"     17.2
ПАО "Юнипро"       4.0
Name: P/E 2022 год, dtype: float64
```

```
1 import pandas as pd
2
3 d1 = {'ПАО "Россети"': 54_210_000_000, 'ПАО "РусГидро"': 19_325_000_000,
4       'ПАО "Юнипро"': 21_267_722_000}
5 srs1 = pd.Series(d1, name='Чистая прибыль за 2022 год, руб.')
6 d2 = {'ПАО "Россети"': 110_844_896_493.56, 'ПАО "РусГидро"': 333_200_635_086.47,
7       'ПАО "Юнипро"': 84_863_558_471.17}
8 srs2 = pd.Series(d2, name='Капитализация на конец 2022 года, в млн рублей')
9 # Продолжите написание кода здесь
```

Имеется список `lst`, содержащий названия компаний, и строка `s` с наименованием показателя. На основе списка `lst` создайте объект `Series`. Затем создайте новую серию, которая будет содержать наименования показателей каждой компании. Выведите результат на печать.

Sample Input:

Sample Output:

```
0          Выручка ПАО "Мосэнерго"
1  Выручка ПАО "ГМК "Норильский никель"
2          Выручка ПАО "Магнит"
3          Выручка ПАО "МТС"
4          Выручка ПАО "ММК"
5          Выручка ПАО "ЛУКОЙЛ"
6          Выручка ПАО "Аэрофлот"
dtype: object
```

```
1 import pandas as pd
2
3 lst = ['ПАО "Мосэнерго"', 'ПАО "ГМК "Норильский никель"', 'ПАО "Магнит"',
4        'ПАО "МТС"', 'ПАО "ММК"', 'ПАО "ЛУКОЙЛ"', 'ПАО "Аэрофлот"']
5 s = 'Выручка'
6 # Продолжите написание кода здесь
```

Частный вкладчик разместил в банке пять депозитов на разные сроки. В серии `amounts` представлены суммы депозитов в рублях. В серии `days` перечислены сроки размещения в днях. Серия `rates` содержит ставки в процентах.

```
0      300000
1      200000
2      500000
3      700000
4     1000000
Name: Сумма депозита, dtype: int64
```

```
0      30
1      60
2      90
3     120
4     180
Name: Дни, dtype: int64
```

```
0      10
1      11
2      11
3      12
4      14
Name: Ставки, dtype: int64
```

Посчитайте суммы процентов, которые получит вкладчик по каждому депозиту. База для расчета процентов = 365 дней. Округлите результат до **двух** знаков после запятой и выведите итоговый объект `Series` на печать.

Sample Input:

Sample Output:

```
0      2465.75
1      3616.44
2     13561.64
3     27616.44
4     69041.10
dtype: float64
```

```
1 import pandas as pd
2
3 amounts = [300_000, 200_000, 500_000, 700_000, 1_000_000]
4 days = [30, 60, 90, 120, 180]
5 rates = [10, 11, 11, 12, 14]
6 amounts = pd.Series(amounts, name='Сумма депозита')
7 days = pd.Series(days, name='Дни')
8 rates = pd.Series(rates, name='Ставки')
9 # Продолжите написание кода здесь
```

Даны два объекта `Series` . Создайте новую серию, которая будет содержать разность дней между значениями серии `srs1` и серии `srs2` . Выведите полученный объект `Series` на печать.

Тип данных в итоговой серии должен быть `'timedelta64[ns]'` .

```
print(srs1)
```

```
0    2023-01-01
1    2023-02-01
2    2023-03-01
3    2023-04-01
4    2023-05-01
5    2023-06-01
6    2023-07-01
7    2023-08-01
dtype: object
```

```
print(srs2)
```

```
0    2020-01-01
1    2020-01-01
2    2020-01-01
3    2020-01-01
4    2020-01-01
5    2020-01-01
6    2020-01-01
7    2020-01-01
dtype: object
```

Sample Input:

Sample Output:

```
0    1096 days
1    1127 days
2    1155 days
3    1186 days
4    1216 days
5    1247 days
6    1277 days
7    1308 days
dtype: timedelta64[ns]
```

```
1 import pandas as pd
2
3 lst1 = ['2023-01-01', '2023-02-01', '2023-03-01', '2023-04-01',
4         '2023-05-01', '2023-06-01', '2023-07-01', '2023-08-01',]
5 srs1 = pd.Series(lst1)
6 srs2 = pd.Series('2020-01-01', index=range(8))
7 # Продолжите написание кода здесь
```

Трейдер выставил биржевые заявки на покупку акций по ценам, представленным в серии `srs1` . Некоторые из этих заявок исполнились, и в серии `srs2` находятся данные по количеству купленных ценных бумаг.

ПАО "Газпром"	162.500
Банк ВТБ (ПАО)	0.024
ПАО "Аэрофлот"	36.500
ПАО "Магнит"	6200.000
ПАО "Газпром нефть"	830.000
ПАО "ЛУКОЙЛ"	7000.000
ПАО "РусГидро"	0.800
ПАО "Северсталь"	1250.000
ПАО "ММК"	50.000
ПАО "Мосэнерго"	2.900
ПАО "МТС"	258.000
Name: Цена в заявке на покупку, руб., dtype: float64	

ПАО "Аэрофлот"	10000
Банк ВТБ (ПАО)	10000000
ПАО "Газпром"	1000
ПАО "Мосэнерго"	10000
ПАО "МТС"	500
ПАО "РусГидро"	300000
Name: Количество купленных акций, шт., dtype: int64	

Выведите на печать объект `Series` , содержащий общие суммы потраченных на покупку денежных средств по каждой компании. По тем акциям, по которым заявки не реализовались, должны быть значения `NaN` .

Sample Input:

Sample Output:

Банк ВТБ (ПАО)	240000.0
ПАО "Аэрофлот"	365000.0
ПАО "Газпром нефть"	NaN
ПАО "Газпром"	162500.0
ПАО "ЛУКОЙЛ"	NaN
ПАО "ММК"	NaN
ПАО "МТС"	129000.0
ПАО "Магнит"	NaN
ПАО "Мосэнерго"	29000.0
ПАО "РусГидро"	240000.0
ПАО "Северсталь"	NaN
dtype: float64	

```
1 import pandas as pd
2
3 d1 = {'ПАО "Газпром"': 162.5, 'Банк ВТБ (ПАО)': 0.024, 'ПАО "Аэрофлот"': 36.5,
4       'ПАО "Магнит"': 6200, 'ПАО "Газпром нефть"': 830, 'ПАО "ЛУКОЙЛ"': 7000,
5       'ПАО "РусГидро"': 0.8, 'ПАО "Северсталь"': 1250, 'ПАО "ММК"': 50,
6       'ПАО "Мосэнерго"': 2.9, 'ПАО "МТС"': 258}
7 d2 = {'ПАО "Аэрофлот"': 10_000, 'Банк ВТБ (ПАО)': 10_000_000, 'ПАО "Газпром"': 1_000,
8       'ПАО "Мосэнерго"': 10_000, 'ПАО "МТС"': 500, 'ПАО "РусГидро"': 300_000}
9 srs1 = pd.Series(d1, name='Цена в заявке на покупку, руб.')
10 srs2 = pd.Series(d2, name='Количество купленных акций, шт.')
11 # Продолжите написание кода здесь
```

Трейдер выставил биржевые заявки на покупку акций по ценам, представленным в серии `srs1` . Некоторые из этих заявок исполнились, и в серии `srs2` находятся данные по количеству купленных ценных бумаг.

ПАО "Газпром"	162.500
Банк ВТБ (ПАО)	0.024
ПАО "Аэрофлот"	36.500
ПАО "Магнит"	6200.000
ПАО "Газпром нефть"	830.000
ПАО "ЛУКОЙЛ"	7000.000
ПАО "РусГидро"	0.800
ПАО "Северсталь"	1250.000
ПАО "ММК"	50.000
ПАО "Мосэнерго"	2.900
ПАО "МТС"	258.000
Name: Цена в заявке на покупку, руб., dtype: float64	

ПАО "Аэрофлот"	10000
Банк ВТБ (ПАО)	100000000
ПАО "Газпром"	1000
ПАО "Мосэнерго"	10000
ПАО "МТС"	500
ПАО "РусГидро"	300000
Name: Количество купленных акций, шт., dtype: int64	

Выведите на печать объект `Series` , содержащий общие суммы потраченных на покупку денежных средств по каждой компании. По тем акциям, по которым заявки не реализовались, должны быть **нулевые** значения.

Sample Input:

Sample Output:

Банк ВТБ (ПАО)	240000.0
ПАО "Аэрофлот"	365000.0
ПАО "Газпром нефть"	0.0
ПАО "Газпром"	162500.0
ПАО "ЛУКОЙЛ"	0.0
ПАО "ММК"	0.0
ПАО "МТС"	129000.0
ПАО "Магнит"	0.0
ПАО "Мосэнерго"	29000.0
ПАО "РусГидро"	240000.0
ПАО "Северсталь"	0.0
dtype: float64	

```
1 import pandas as pd
2
3 d1 = {'ПАО "Газпром"': 162.5, 'Банк ВТБ (ПАО)': 0.024, 'ПАО "Аэрофлот"': 36.5,
4       'ПАО "Магнит"': 6200, 'ПАО "Газпром нефть"': 830, 'ПАО "ЛУКОЙЛ"': 7000,
5       'ПАО "РусГидро"': 0.8, 'ПАО "Северсталь"': 1250, 'ПАО "ММК"': 50,
6       'ПАО "Мосэнерго"': 2.9, 'ПАО "МТС"': 258}
7 d2 = {'ПАО "Аэрофлот"': 10_000, 'Банк ВТБ (ПАО)': 10_000_000, 'ПАО "Газпром"': 1_000,
8       'ПАО "Мосэнерго"': 10_000, 'ПАО "МТС"': 500, 'ПАО "РусГидро"': 300_000}
9 srs1 = pd.Series(d1, name='Цена в заявке на покупку, руб.')
10 srs2 = pd.Series(d2, name='Количество купленных акций, шт.')
11 # Продолжите написание кода здесь
```

Треjder выставил биржевые заявки на покупку акций по ценам, представленным в серии `srs1` . Некоторые из этих заявок исполнились, и в серии `srs2` находятся данные по количеству купленных ценных бумаг.

```
ПАО "Газпром"          162.500
Банк ВТБ (ПАО)         0.024
ПАО "Аэрофлот"         36.500
ПАО "Магнит"           6200.000
ПАО "Газпром нефть"    830.000
ПАО "ЛУКОЙЛ"           7000.000
ПАО "РусГидро"         0.800
ПАО "Северсталь"       1250.000
ПАО "ММК"              50.000
ПАО "Мосэнерго"        2.900
ПАО "МТС"              258.000
Name: Цена в заявке на покупку, руб., dtype: float64
```

```
ПАО "Газпром"          1000
Банк ВТБ (ПАО)       10000000
ПАО "Аэрофлот"       10000
ПАО "Магнит"          0
ПАО "Газпром нефть"   0
ПАО "ЛУКОЙЛ"          0
ПАО "РусГидро"       300000
ПАО "Северсталь"      0
ПАО "ММК"             0
ПАО "Мосэнерго"      10000
ПАО "МТС"             500
Name: Количество купленных акций, шт., dtype: int64
```

Умножьте серию `srs1` на `srs2` с использованием `*` и выведите полученный объект `Series` на печать.

Обратите внимание на разницу в полученных данных с предыдущей задачей.

Sample Input:

Sample Output:

```
ПАО "Газпром"          162500.0
Банк ВТБ (ПАО)       240000.0
ПАО "Аэрофлот"       365000.0
ПАО "Магнит"          0.0
ПАО "Газпром нефть"   0.0
ПАО "ЛУКОЙЛ"          0.0
ПАО "РусГидро"       240000.0
ПАО "Северсталь"      0.0
ПАО "ММК"             0.0
ПАО "Мосэнерго"      29000.0
ПАО "МТС"            129000.0
dtype: float64
```

```
1 import pandas as pd
2
3 d1 = {'ПАО "Газпром"': 162.5, 'Банк ВТБ (ПАО)': 0.024, 'ПАО "Аэрофлот"': 36.5,
4       'ПАО "Магнит"': 6200, 'ПАО "Газпром нефть"': 830, 'ПАО "ЛУКОЙЛ"': 7000,
5       'ПАО "РусГидро"': 0.8, 'ПАО "Северсталь"': 1250, 'ПАО "ММК"': 50,
6       'ПАО "Мосэнерго"': 2.9, 'ПАО "МТС"': 258}
7 d2 = {'ПАО "Газпром"': 1_000, 'Банк ВТБ (ПАО)': 10_000_000, 'ПАО "Аэрофлот"': 10_000,
8       'ПАО "Магнит"': 0, 'ПАО "Газпром нефть"': 0, 'ПАО "ЛУКОЙЛ"': 0,
9       'ПАО "РусГидро"': 300_000, 'ПАО "Северсталь"': 0, 'ПАО "ММК"': 0,
10      'ПАО "Мосэнерго"': 10_000, 'ПАО "МТС"': 500}
11 srs1 = pd.Series(d1, name='Цена в заявке на покупку, руб.')
12 srs2 = pd.Series(d2, name='Количество купленных акций, шт.')
13 # Продолжите написание кода здесь
```

На вход подается строка, содержащая шесть натуральных чисел, разделенных пробелами.

- На основе этих чисел создайте объект `Series` .
- Затем возведите в степень значения этой серии таким образом, чтобы первое значение возвелось в первую степень, второе значение - во вторую степень, третье - в третью, четвертое - в четвертую, пятое - в пятую и шестое значение - в шестую степень.
- Выведите полученный объект `Series` на печать.

Sample Input:

1 2 3 3 2 1

Sample Output:

```
0      1
1      4
2     27
3     81
4     32
5      1
dtype: int64
```