In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from pylab import plot
```

In [2]:
```
# Подключаем и читаем файл с данными metal.csv
values_df = pd.read_csv('metal.csv', delimiter=';')
```

In [3]:
```
# получение типов данных столбцов
print(values_df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 682 entries, 0 to 681
Data columns (total 5 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   Date       682 non-null    object 
 1   Gold       682 non-null    float64
 2   Silver     682 non-null    float64
 3   Platinum   682 non-null    float64
 4   Palladium  682 non-null    float64
dtypes: float64(4), object(1)
memory usage: 26.8+ KB
None
```

In [4]:
```
# Вывод названий столбцов таблицы
print(values_df.columns)
```

```
Index(['Date', 'Gold', 'Silver', 'Platinum', 'Palladium'], dtype='object')
```

In [5]:
```
# Вывод данных таблицы
print(values_df.head())
```

```
              Date     Gold  Silver  Platinum  Palladium
0  13.10.2021 0:00  4065.28   52.32   2343.35    4897.00
1  12.10.2021 0:00  4036.46   51.87   2381.50    5007.14
2  09.10.2021 0:00  4067.69   52.19   2295.96    4619.69
3  08.10.2021 0:00  4086.92   52.73   2291.49    4420.30
4  07.10.2021 0:00  4078.88   52.02   2209.46    4418.93
```

In [6]:
```
# преобразование данных в массив
# приведение типа DataFrame к типу ndArray и выбор 2-го столбца
values_arr2 = np.array(values_df)[:, 2]
print(values_arr2)
```

```
[52.32 51.87 52.19 52.73 52.02 52.46 52.31 51.8 50.29 52.23 51.75 52.69
 53.19 53.17 53.27 53.09 52.8 53.68 54.66 55.83 55.17 55.41 56.2 56.77
 57.41 57.05 57.95 56.34 56.6 56.29 56.65 55.85 56.16 56.32 56.32 56.49
 55.79 55.48 55.64 55.91 56.46 55.55 55.24 55.38 55.68 55.34 56.34 58.72
 59.87 60.07 59.65 59.64 59.94 59.89 58.68 59.63 60.43 59.68 59.4 60.13
 60.1 60.53 62.25 62.53 62.37 62.16 62.21 62.27 63.11 62.82 62.68 62.6
 61.94 61.56 60.24 60.39 60.54 60.63 60.44 60.62 61.09 61.08 61.27 62.0
 64.17 63.8 64.85 64.2 63.92 64.89 64.67 64.47 65.38 65.56 66.43 65.11
 65.37 65.31 66.43 65.02 65.35 65.77 65.64 65.56 67.47 65.56 64.78 64.12
 65.25 65.52 65.25 64.08 63.29 64.9 62.26 62.86 62.33 63.35 62.76 63.07
 64.62 63.99 63.25 64.0 63.5 63.32 61.63 61.97 62.7 62.58 62.79 62.47
```

```
61.49 59.89 59.47 59.26 58.36 59.24 60.53 60.09 61.7 62.73 62.36 62.79
62.28 61.22 61.3 61.22 59.98 62.04 60.6 61.31 60.06 62.11 63.07 62.07
64.47 63.53 66.83 65.54 66.31 64.1 64.6 64.32 65.4 64.96 63.8 64.48 64.57
65.62 64.77 63.34 63.75 65.25 66.84 72.2 66.55 61.79 61.69 61.28 62.34
60.94 61.84 59.67 59.66 58.94 60.04 59.7 60.11 60.34 59.74 63.83 62.48
62.11 61.05 61.1 61.08 60.91 62.52 63.36 61.98 60.67 58.96 57.16 56.33
55.84 56.15 57.08 57.68 56.25 57.83 57.7 57.98 57.22 54.35 56.68 56.97
56.79 56.19 58.27 58.86 58.6 59.62 60.12 59.49 59.96 60.2 60.02 59.28
61.66 63.78 60.97 60.47 61.67 61.2 58.7 60.67 60.93 59.68 60.85 60.87
61.74 61.97 61.59 62.23 61.06 59.98 60.06 61.96 62.06 60.22 60.02 59.08
61.34 60.07 59.92 58.68 60.1 61.06 57.9 55.89 55.14 57.92 59.59 64.44
65.28 64.66 65.96 66.59 64.58 64.71 65.82 64.9 65.15 65.21 64.82 65.27
65.79 68.34 64.9 65.63 65.9 64.1 63.47 64.33 63.97 63.84 64.61 66.87
63.14 62.86 62.12 60.77 66.51 67.05 67.07 65.48 63.48 57.4 57.77 56.82
54.63 56.52 54.44 56.1 51.74 52.16 50.63 46.9 44.98 44.18 44.07 44.0
43.21 43.43 42.98 42.92 42.34 41.81 41.87 40.84 40.64 41.26 40.41 40.27
39.63 39.15 39.52 39.42 39.9 39.2 39.43 39.03 39.04 38.67 39.72 39.08
38.93 38.72 38.79 39.39 39.24 40.52 40.7 40.01 39.64 39.27 39.57 39.13
39.29 39.23 40.63 39.85 40.71 38.25 36.91 36.81 36.56 35.58 35.7 35.85
35.86 35.93 36.31 36.42 36.79 36.97 36.93 36.7 36.37 36.04 37.23 36.9
36.56 35.87 35.98 36.4 36.69 36.57 35.76 36.03 35.35 34.08 31.71 32.84
30.12 32.01 30.88 30.8 37.58 38.88 40.51 39.22 39.1 37.34 36.71 35.71
36.13 36.65 38.88 38.13 38.61 39.19 38.01 37.59 36.62 36.49 36.04 35.99
35.91 35.89 36.57 36.43 36.25 35.57 36.01 36.24 36.75 35.98 35.41 36.06
36.96 35.74 34.97 35.39 35.75 35.91 35.69 35.63 35.33 35.1 35.51 35.1
35.28 36.26 35.48 35.48 35.07 34.92 34.89 34.73 34.09 33.99 34.17 34.33
34.22 34.18 33.92 33.87 34.08 33.97 34.71 34.56 35.14 35.11 34.8 35.13
34.91 35.15 34.83 34.73 35.21 35.01 35.13 35.25 34.44 34.58 34.99 34.99
34.54 34.65 34.54 35.92 35.94 36.89 36.98 37.17 36.82 36.27 37.05 37.22
37.29 35.99 35.97 36.06 36.34 35.85 35.91 35.7 36.5 36.43 36.34 36.95
37.21 36.61 36.32 36.81 36.85 36.31 35.79 35.86 36.28 37.14 38.32 38.01
37.79 36.7 36.66 36.98 36.78 36.6 37.61 38.13 38.21 37.85 38.3 38.5 40.82
41.27 39.74 39.31 39.3 39.66 39.21 37.76 36.16 35.96 35.78 36.45 36.54
36.24 36.41 36.63 35.88 36.79 35.67 35.67 35.64 35.2 34.35 34.48 33.66
32.85 33.59 33.53 33.47 33.37 33.58 33.55 33.35 33.18 32.98 32.38 31.59
31.09 30.95 30.68 30.92 30.96 30.8 30.95 31.09 31.15 31.25 30.84 30.94
30.87 30.91 30.8 30.96 31.09 31.03 31.24 30.74 30.81 30.56 31.13 30.74
30.48 30.72 31.18 31.37 31.07 30.84 30.99 30.44 30.12 30.0 30.05 30.16
30.22 30.02 29.9 29.94 29.89 30.09 30.71 30.89 30.98 30.86 31.31 31.09
30.78 31.14 31.14 31.2 30.91 30.45 30.71 30.67 30.75 30.81 30.84 30.87
30.85 31.25 31.39 31.73 31.77 31.81 31.95 31.68 31.77 31.63 31.7 31.62
32.08 32.06 32.03 32.06 31.86 31.4 31.85 31.8 31.93 32.04 32.63 32.57
32.34 32.11 32.01 31.99 32.07 32.92 33.46 33.6 33.47 33.66 33.3 33.51
33.79 33.41 33.55 33.39 33.41 33.58 33.38 33.18 33.31 33.37 33.39 33.5
33.19 33.76 33.93 33.44 33.69 33.45 32.59 32.42 32.65 32.54 32.65 33.01
33.22 33.2 33.48 33.68 33.87 33.79 33.58 33.73]
```

In [7]:

```
# приведение типа DataFrame к типу ndArray и выбор 2-го и 3-го столбца
values_arr23 = np.array(values_df)[:, 2:4]
print(values_arr23)
```

```
[[52.32 2343.35]
 [51.87 2381.5]
 [52.19 2295.96]
 ...
 [33.79 1770.62]
 [33.58 1771.28]
 [33.73 1774.93]]
```

In [8]:

```
# визуализация данных
plt.figure()
x = np.linspace(values_arr2[0], values_arr23[values_arr2.size - 1], values_arr2.size)
```

```
plot(x, values_arr2, 'r')
plt.show()
```