



# BÀI TẬP THỰC HÀNH

Bài 4:

LẬP TRÌNH PYTHON CƠ BẢN

(Sử dụng thư viện NumPy làm việc với ma trận - 01)



**VINBIGDATA** **VINGROUP**



# Thực hành 1



**Yêu cầu 1: Học viên tạo một ma trận vuông cấp  $n$  ( $n$  hàng x  $n$  cột), bao gồm các phần tử là những số nguyên ngẫu nhiên trong khoảng (0-100) như minh họa, với  $n = 12$**

```
[ [61 21 68 72 84 90 81 80 79 80 49 53]
  [92 10  1 43 49 93 76  6  2  6 69  2]
  [12 79 88 10 37 55 37  6 59 75 77 64]
  [97 22 75 32 39 39 93 19 28 64 55 87]
  [88 25 88 58 11 96 58 14 88 16 22 64]
  [ 3  5 60 14 65 50 80 42  8 27 44 52]
  [84 38 54 27 86 13 67 77 77 12 66 40]
  [96  9 94 24 61 19  2 80 95 92 72 32]
  [49 21 78 92 35 92 84 86 85 62 64 29]
  [ 5 53 95  2 43 30 72 66 97 17  8 23]
  [16 27  1 71 19 22 90 81 12 93 14 53]
  [40 50 83 25 37 16 49 73 42 86 11 18]]
```

Kiểu dữ liệu của phần tử trong ma trận: int32

Kích thước của mảng ma trận: (12, 12)

Số phần tử của mảng ma trận: 144

Số chiều của mảng ma trận: 2



**Yêu cầu 2: Sử dụng ma trận tạo được trong yêu cầu 1, Học viên tạo 2 vector như sau:**

- **v\_chinh:** bao gồm các phần tử nằm trên đường chéo chính của ma trận.
- **V\_phu:** bao gồm các phần tử nằm trên đường chéo phụ của ma trận

```
[[61 21 68 72 84 90 81 80 79 80 49 53]
 [92 10 1 43 49 93 76 6 2 6 69 2]
 [12 79 88 10 37 55 37 6 59 75 77 64]
 [97 22 75 32 39 39 93 19 28 64 55 87]
 [88 25 88 58 11 96 58 14 88 16 22 64]
 [ 3  5 60 14 65 50 80 42  8 27 44 52]
 [84 38 54 27 86 13 67 77 77 12 66 40]
 [96  9 94 24 61 19  2 80 95 92 72 32]
 [49 21 78 92 35 92 84 86 85 62 64 29]
 [ 5 53 95  2 43 30 72 66 97 17  8 23]
 [16 27  1 71 19 22 90 81 12 93 14 53]
 [40 50 83 25 37 16 49 73 42 86 11 18]]
```

Vector các phần tử nằm trên đường chéo chính:  
[61. 10. 88. 32. 11. 50. 67. 80. 85. 17. 14. 18.]

-----  
Vector các phần tử nằm trên đường chéo phụ:  
[40. 27. 95. 92. 61. 13. 80. 14. 28. 75. 69. 53.]



**Yêu cầu 3: Nhập vào số nguyên x bất kỳ trong khoảng (0-100), đếm xem có bao nhiêu phần tử trong ma trận sinh ra ở yêu cầu 1 có giá trị bằng, lớn hơn và nhỏ hơn giá trị x:**

```
[[61 21 68 72 84 90 81 80 79 80 49 53]
 [92 10 1 43 49 93 76 6 2 6 69 2]
 [12 79 88 10 37 55 37 6 59 75 77 64]
 [97 22 75 32 39 39 93 19 28 64 55 87]
 [88 25 88 58 11 96 58 14 88 16 22 64]
 [ 3  5 60 14 65 50 80 42  8 27 44 52]
 [84 38 54 27 86 13 67 77 77 12 66 40]
 [96  9 94 24 61 19  2 80 95 92 72 32]
 [49 21 78 92 35 92 84 86 85 62 64 29]
 [ 5 53 95  2 43 30 72 66 97 17  8 23]
 [16 27  1 71 19 22 90 81 12 93 14 53]
 [40 50 83 25 37 16 49 73 42 86 11 18]]
```

Nhập vào giá trị x (0-100):88

- 
1. Số phần tử có giá trị bằng x trong ma trận: 4
  2. Số phần tử nhỏ hơn giá trị x trong ma trận: 124
  3. Số phần tử lớn hơn giá trị x trong ma trận: 16



# Thực hành 2



**Yêu cầu 1: Sử dụng dữ liệu bảng điểm của lớp 2A. Cho biết:**

- 1. ĐTB của từng học sinh trong lớp.**
- 2. Học sinh có điểm TB cao nhất.**
- 3. Học sinh có điểm trung bình thấp nhất**

Điểm TB của từng học sinh trong lớp:

```
[4.8 6.  4.7 8.6 6.2 6.8 5.6 5.4 5.9 5.1 7.2 5.4 5.9 5.9 6.  7.9 4.2 6.1  
6.3 4.4 4.7 5.9 5.6 4.7 6.4 6.2 6.5 4.6 5.8 4.3]
```

-----  
Điểm TB cao nhất: 8.6

Của học sinh thứ: 3

Bảng điểm đầy đủ của học sinh: [ 7 10 9 8 7 10 10 8 9 8]

-----  
Điểm TB thấp nhất: 4.2

Của học sinh thứ: 16

Bảng điểm đầy đủ của học sinh: [3 2 2 1 2 6 2 7 9 8]



**Yêu cầu 2: Sử dụng dữ liệu bảng điểm của lớp 2A. Cho biết:**

- 1. ĐTB của từng môn học.**
- 2. Môn học có điểm TB cao nhất.**
- 3. Môn học có điểm trung bình thấp nhất**

Điểm TB của từng môn học:

```
[4.73 4.43 5.5 4.83 4.97 5.6 6.23 7.3 6.13 7.97]
```

Điểm TB cao nhất của môn học: 7.97

Môn học thứ [9]

Bảng điểm đầy đủ của môn học: 

```
[[ 7 8 7 8 6 10 10 6 8 10 8 9 8 8 5 10 8 7 8 7 9 9 8 7
  7 7 10 8 9 7]]
```

Điểm TB thấp nhất của môn học: 4.43

Môn học thứ [1]

Bảng điểm đầy đủ của môn học: 

```
[[ 3 5 3 10 9 1 9 8 3 1 6 0 7 10 8 5 2 7 7 1 1 6 1 6
  3 0 2 2 1 6]]
```



## Yêu cầu 3: Sử dụng dữ liệu bảng điểm của lớp 2A. Cho biết:

1. Sinh viên có điểm đồng đều nhất tất cả các môn. Sinh viên có điểm các môn lệch nhất trong lớp.
2. Môn học có điểm đồng đều nhất. Môn học có điểm chênh lệch nhất.

Học sinh học đồng đều nhất: [3]

Bảng điểm của học sinh học đồng đều: `[[ 7 10 9 8 7 10 10 8 9 8]]`

-----

Học sinh học lệch nhất: [19]

Bảng điểm của học sinh học lệch: `[[ 5 1 0 10 4 7 1 8 1 7]]`

-----

Môn học có điểm đồng đều nhất: [7]

Bảng điểm của môn học đồng đều: `[[8 8 7 8 6 7 7 8 6 7 8 6 7 6 8 8 7 6 8 8 8 7 8 8 8 6 8 7 7 8]]`

-----

Môn học có điểm lệch nhất: [2]

Bảng điểm của môn học lệch: `[[ 1 10 4 9 6 9 0 2 3 1 8 6 8 4 2 9 2 9 5 0 4 1 7 3  
8 9 8 9 9 9]]`



**VINBIGDATA** **VINGROUP**



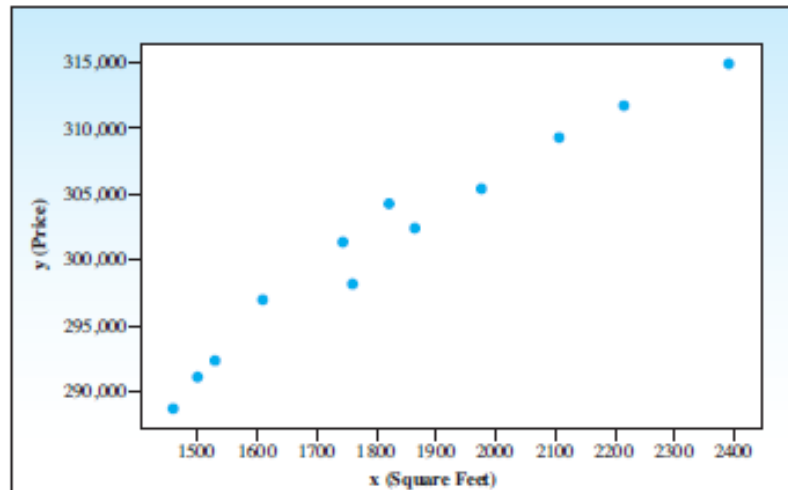
# Thực hành 3



**Yêu cầu: Xác định hệ số tương quan giữa diện tích (1) | Khoảng cách từ trung tâm thành phố (2) và giá bán nhà theo bảng số liệu dưới đây:**

Square Feet, x	Price, y	Square Feet, x	Price, y
1460	\$288,700	1977	\$305,400
2108	309,300	1610	297,000
1743	301,400	1530	292,400
1499	291,100	1759	298,200
1864	302,400	1821	304,300
2391	314,900	2216	311,700

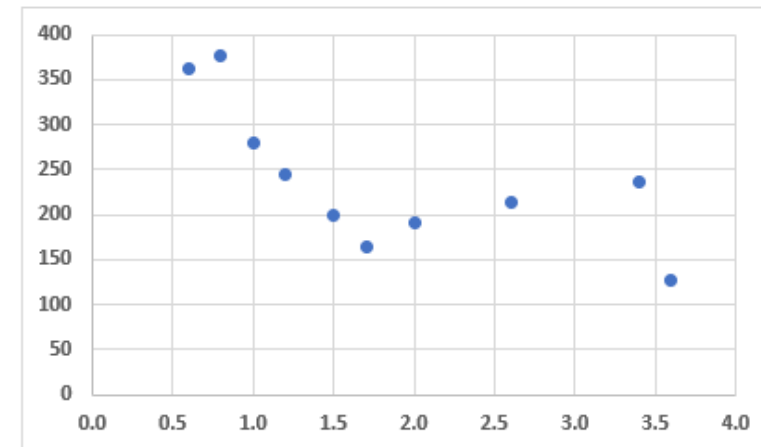
Plot of data for Exercise 12.42



01

Distance (in Miles from the city centre)	Prices (in \$ x 1000)
2.6	214
0.8	376
1.0	280
0.6	362
1.5	200
2.0	190
3.4	236
1.2	244
3.6	128
1.7	165

02



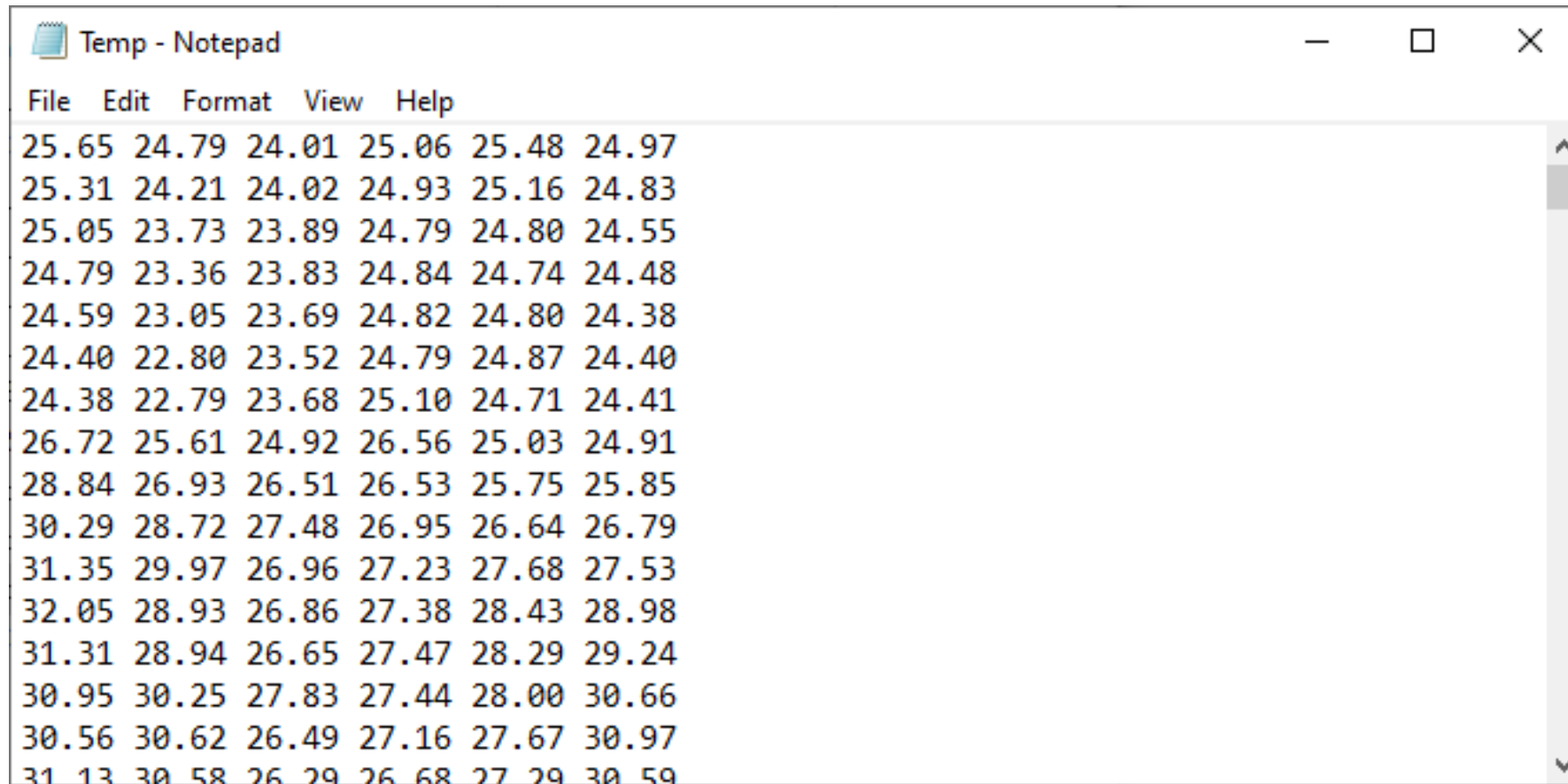


**VINBIGDATA** **VINGROUP**



# Thực hành 4

## Mô tả file dữ liệu: Temp.txt



File	Edit	Format	View	Help	
25.65	24.79	24.01	25.06	25.48	24.97
25.31	24.21	24.02	24.93	25.16	24.83
25.05	23.73	23.89	24.79	24.80	24.55
24.79	23.36	23.83	24.84	24.74	24.48
24.59	23.05	23.69	24.82	24.80	24.38
24.40	22.80	23.52	24.79	24.87	24.40
24.38	22.79	23.68	25.10	24.71	24.41
26.72	25.61	24.92	26.56	25.03	24.91
28.84	26.93	26.51	26.53	25.75	25.85
30.29	28.72	27.48	26.95	26.64	26.79
31.35	29.97	26.96	27.23	27.68	27.53
32.05	28.93	26.86	27.38	28.43	28.98
31.31	28.94	26.65	27.47	28.29	29.24
30.95	30.25	27.83	27.44	28.00	30.66
30.56	30.62	26.49	27.16	27.67	30.97
31.13	30.58	26.29	26.68	27.29	30.59

# Thực hành 4

## Mô tả file dữ liệu: Temp.txt

- File dữ liệu lưu trữ nhiệt độ (°C) của 6 thành phố lớn dọc theo nước Việt Nam là: Hà Nội, Vinh, Đà Nẵng, Nha trang, Hồ Chính Minh và Cà Mau
- Thời gian từ 0h ngày 15/09/2019 tới 23h ngày 22/09/2019



# Thực hành 4

Mô tả file dữ liệu: Temp.txt

Hà Nội	Vinh	Đà Nẵng	Nha Trang	HCM	Cà Mau	
25.65	24.79	24.01	25.06	25.48	24.97	Time: 0h 15/09
25.31	24.21	24.02	24.93	25.16	24.83	1h 15/09
25.05	23.73	23.89	24.79	24.80	24.55	
24.79	23.36	23.83	24.84	24.74	24.48	
24.59	23.05	23.69	24.82	24.80	24.38	
24.40	22.80	23.52	24.79	24.87	24.40	
24.38	22.79	23.68	25.10	24.71	24.41	
26.72	25.61	24.92	26.56	25.03	24.91	
28.84	26.93	26.51	26.53	25.75	25.85	
30.29	28.72	27.48	26.95	26.64	26.79	

Yêu cầu 1) Đọc dữ liệu lưu trữ trong file Temp.txt vào biến data\_numpy, cho biết kích thước, số chiều, kiểu dữ liệu và số phần tử của biến data\_numpy.

```
1 print(data_numpy)
2 print('-----')
3 print('Kích thước biến:', data_numpy.shape)
4 print('Số chiều của biến:', data_numpy.ndim)
5 print('Kiểu dữ liệu của các phần tử:', data_numpy.dtype)
6 print('Số phần tử:', data_numpy.size)
```

```
[[25.65 24.79 24.01 25.06 25.48 24.97]
 [25.31 24.21 24.02 24.93 25.16 24.83]
 [25.05 23.73 23.89 24.79 24.8  24.55]
 ...
 [24.81 24.47 23.4  25.86 25.05 25.29]
 [23.97 24.22 22.95 25.74 24.92 24.87]
 [22.84 23.99 22.59 25.5  24.77 24.57]]
```

```
-----
Kích thước biến: (192, 6)
Số chiều của biến: 2
Kiểu dữ liệu của các phần tử: float64
Số phần tử: 1152
```



Yêu cầu 2) Tìm nhiệt độ cao nhất (Max) – Thấp nhất (Min) – Nhiệt độ trung bình của cả 6 thành phố.

Yêu cầu 3) Tìm nhiệt độ cao nhất (Max) – Thấp nhất (Min) – Nhiệt độ trung bình của từng thành phố và hiển thị kết quả.

---THÔNG KÊ CHO CẢ 6 THÀNH PHỐ---

Nhiệt độ cao nhất: 33.45

Nhiệt độ thấp nhất: 20.93

Nhiệt độ trung bình: 26.50222222222222

-----  
1) Hà Nội

Nhiệt độ cao nhất: 33.45

Nhiệt độ thấp nhất: 21.68

Nhiệt độ trung bình: 27.71229166666667

2) Vinh (Nghệ An)

Nhiệt độ cao nhất: 32.57

Nhiệt độ thấp nhất: 22.6

Nhiệt độ trung bình: 26.719895833333336

3) Đà Nẵng

Nhiệt độ cao nhất: 29.88

Nhiệt độ thấp nhất: 20.93

Nhiệt độ trung bình: 25.522499999999997

4) Nha Trang

Nhiệt độ cao nhất: 28.68

Nhiệt độ thấp nhất: 24.5

Nhiệt độ trung bình: 26.166875000000005

5) TP Hồ Chí Minh

Nhiệt độ cao nhất: 31.06

Nhiệt độ thấp nhất: 23.22

Nhiệt độ trung bình: 26.159218749999997

6) Cà Mau

Nhiệt độ cao nhất: 31.37

Nhiệt độ thấp nhất: 23.99

Nhiệt độ trung bình: 26.732552083333333

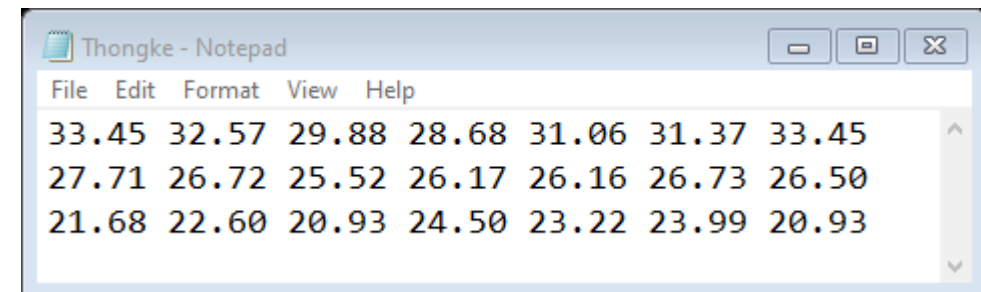
Yêu cầu 4) Tạo một ma trận **data\_thongke** gồm 3 hàng x 7 cột; các hàng lần lượt lưu trữ dữ liệu như sau:

- hàng 0: Nhiệt độ lớn nhất (Max)
- hàng 1: Nhiệt độ trung bình (Mean), làm tròn đến 2 số sau dấu phẩy
- hàng 2: Nhiệt độ nhỏ nhất (Min)

Các cột lần lượt theo thứ tự của 6 thành phố và cột cuối cùng là cột thống kê chung cho cả 6 thành phố. Lưu ra file **thongke.txt**

```
1 print(data_thongke)
2 print(type(data_thongke))
3 print('Kích thước:', data_thongke.shape)

[[33.45 32.57 29.88 28.68 31.06 31.37 33.45]
 [27.71 26.72 25.52 26.17 26.16 26.73 26.5 ]
 [21.68 22.6  20.93 24.5  23.22 23.99 20.93]]
<class 'numpy.ndarray'>
Kích thước: (3, 7)
```



Thongke - Notepad

File Edit Format View Help

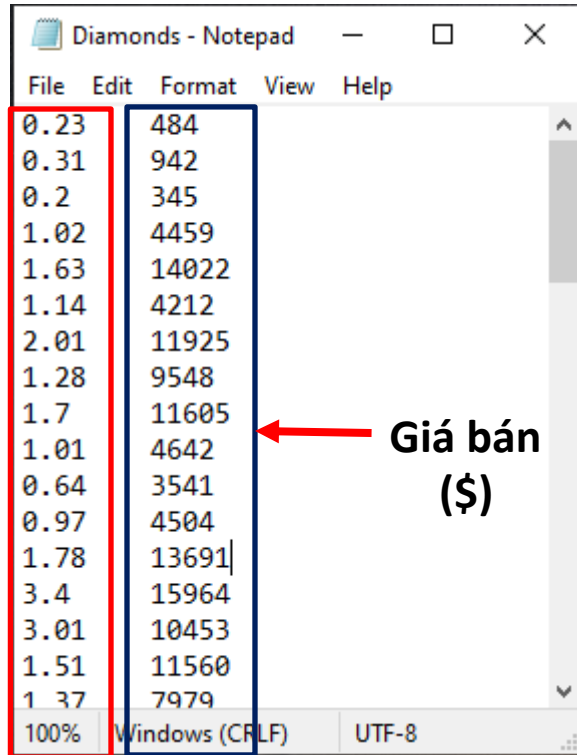
33.45 32.57 29.88 28.68 31.06 31.37 33.45  
27.71 26.72 25.52 26.17 26.16 26.73 26.50  
21.68 22.60 20.93 24.50 23.22 23.99 20.93



# Thực hành 5

# Thực hành 5

## File dữ liệu: Diamonds.txt



0.23	484
0.31	942
0.2	345
1.02	4459
1.63	14022
1.14	4212
2.01	11925
1.28	9548
1.7	11605
1.01	4642
0.64	3541
0.97	4504
1.78	13691
3.4	15964
3.01	10453
1.51	11560
1.37	7979

Trọng lượng  
(carat)

Giá bán  
(\$)

### Mô tả file dữ liệu:

- File dữ liệu lưu trữ thông số 50 viên kim cương bao gồm: Trọng lượng (carat) và Giá bán (\$) tương ứng



## Học viên thực hiện các yêu cầu sau:

Yêu cầu 1) Đọc dữ liệu lưu trữ trong file Diamonds.txt vào biến kiểu mảng **data\_diamond**, cho biết kích thước, số chiều, kiểu dữ liệu và số phần tử của biến data\_diamond

```
[3.4000e-01 7.6500e+02]  
[4.1000e-01 8.2700e+02]  
[7.5000e-01 3.1200e+03]  
[1.0700e+00 5.2200e+03]  
[1.3400e+00 7.4270e+03]  
[1.7500e+00 9.8900e+03]]
```

```
-----  
Kích thước biến data_diamond: (50, 2)  
Số chiều của biến data_diamond: 2  
Kiểu dữ liệu của các phần tử: float64  
Số phần tử: 100
```

**Yêu cầu 2) Tách mảng data\_diamond thành 2 vector: diamond\_size và diamond\_price lưu trữ trọng lượng và giá bán.**

Vector diamond\_size:

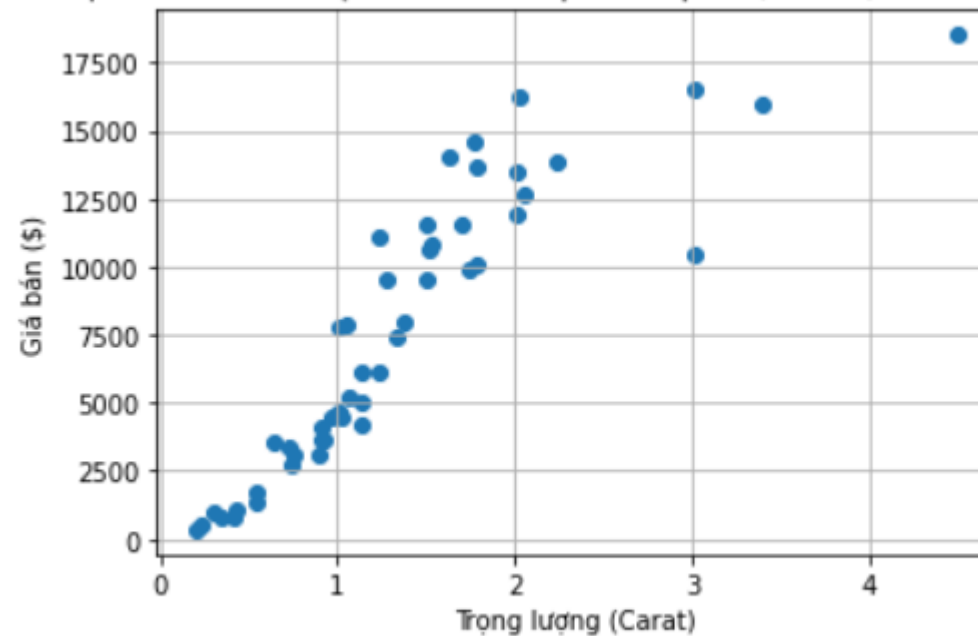
```
[0.23 0.31 0.2 1.02 1.63 1.14 2.01 1.28 1.7 1.01 0.64 0.97 1.78 3.4  
3.01 1.51 1.37 1.5 0.54 0.72 1.13 2.24 3.01 4.5 0.92 1.05 0.55 0.74  
0.91 1.23 1.52 0.91 0.43 1.24 1.77 1.79 2.05 2.03 2.01 1. 0.9 1.01  
1.14 1.53 0.34 0.41 0.75 1.07 1.34 1.75]
```

Vector diamond\_price:

```
[ 484. 942. 345. 4459. 14022. 4212. 11925. 9548. 11605. 4642.  
3541. 4504. 13691. 15964. 10453. 11560. 7979. 9533. 1723. 3344.  
6133. 13827. 16538. 18531. 3625. 7879. 1319. 2761. 3620. 6165.  
10640. 4138. 1094. 11130. 14561. 10108. 12654. 16280. 13498. 4586.  
3105. 7745. 5047. 10830. 765. 827. 3120. 5220. 7427. 9890.]
```

Yêu cầu 3) Vẽ đồ thị thể hiện mối quan hệ giữa kích thước và giá bán kim cương. Xác định hệ số tương quan tương ứng giữa 2 thông số này.

BIỂU ĐỒ THỂ HIỆN MỐI TƯƠNG QUAN GIỮA TRỌNG LƯỢNG (CARAT) VÀ GIÁ BÁN KIM CƯƠNG (\$)



Hệ số tương quan giữa trọng lượng và giá bán kim cương: 0.8814849023922127



Yêu cầu 4) Cho biết kích thước và giá trung bình của 50 viên kim cương. Hiển thị giá bán của viên kim cương có trọng lượng 3.01 carat.

```
Trọng lượng trung bình: 1.3448  
Giá trung bình: 7550.78
```

```
Viên kim cương trọng lượng 3.01 carat có giá bán:  
Giá bán 1 : 10453.0  
Giá bán 2 : 16538.0
```



# Thank you!