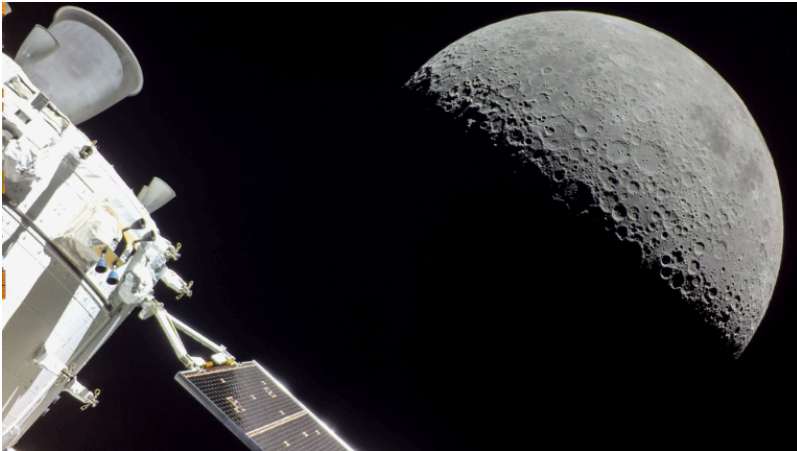


Представим, что космический аппарат, пролетая Луну, фотографирует её на камеру:

(картинка: <https://cdn.mos.cms.futurecdn.net/9iGp27oCPGgEPRFk4zJHDJ.jpg>)



Специальный алгоритм на картинке выделяет на картинке несколько точек на окружности края Луны. Поскольку половина Луны в тени, точки также известны только с одной половины. Координаты (x_i, y_i) для i -й точки следующие:

```
x = [18, 12, 9, 9, 12, 18, 27, 37, 47, 57]
y = [56, 65, 74, 85, 94, 103, 108, 111, 110, 106]
```

Вы можете отобразить эти координаты, чтобы посмотреть, где они находятся:

```
import matplotlib.pyplot as plt
plt.scatter(x, y)
plt.axis('equal')
plt.grid()
plt.show()
```

Как найти центр Луны на картинке по этим точкам? Мы знаем, что центр (x_0, y_0) равноудалён от них, то есть

$$(x_1 - x_0)^2 + (y_1 - y_0)^2 = \dots = (x_N - x_0)^2 + (y_N - y_0)^2,$$

что можно записать короче:

$$\Delta_1 = \dots = \Delta_N.$$

Фактически, координаты точек (x_i, y_i) измерены с некоторой погрешностью, поэтому строгого равенства не будет. Было бы хорошо, если бы такие расстояния Δ_i от центра (x_0, y_0) до точек на окружности (x_i, y_i) были максимально близки друг к другу. Например, попарно: $\Delta_1 \approx \dots \approx \Delta_N$. Тогда можно оценить центр Луны (x_0, y_0) как некоторую точку, для которой попарная разность расстояний минимальна. Эту величину мы назовём ошибкой E (error), и будем искать её наименьшее значение (min):

$$E = (\Delta_1 - \Delta_2)^2 + \dots + (\Delta_{N-1} - \Delta_N)^2 + (\Delta_N - \Delta_1)^2 \rightarrow \min.$$

Найдите на диапазоне координат $x \in [1, 150]$, $y \in [1, 150]$ с шагом 1 (пиксель) наименьшее значение ошибки. Найдите координаты x, y с наименьшим значением ошибки E .

Запишите в двумерный массив E , затем следующим образом изобразите на графике:

```
import matplotlib.pyplot as plt
import numpy as np

fig, ax = plt.subplots(subplot_kw={"projection": "3d"})
X, Y = np.meshgrid(range(1,151), range(1,151))
ax.plot_surface(X, Y, np.array(E))
plt.show()
```