Q1. Mean, var, std manually

arr = [2, 7, 3, 12, 9]

mean = (arr[0] + arr[1] + arr[2] + arr[3] + arr[4]) / len(arr)

variance = ((arr[0] - mean) ** 2 + (arr[1] - mean) ** 2 … (arr[n] - mean) ** 2) / len(arr)

std = …

Q2. Find Euclidean distance between given two points

P = (3, 2)
Q = (4, 1)

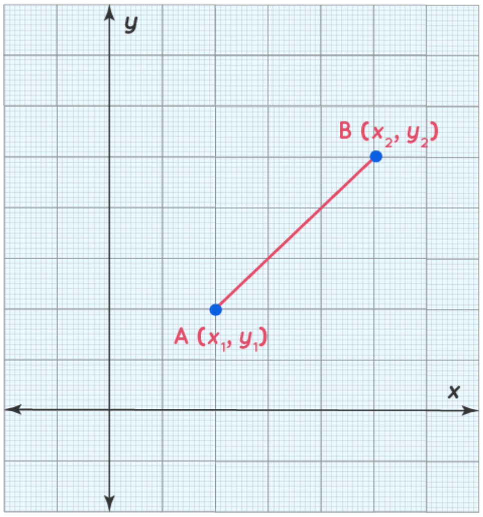
```
Hint: p1 = [3, 2], p2 = [4, 1]
           x1, y1      x2, y2

x1 = p1[0], x2 = p2[0]
```

Nida - https://replit.com/@NidaNida4/PowerfulHeavyDegrees#main.py
Kumayl - https://replit.com/@KumailHussain/SlipperyLegitimateLint#main.py
Syed Ahsan - https://replit.com/@SyedAhsan7/class#main.py
Muslim - https://replit.com/@ratherhussain/FirstCaringAgents

Euclidean Distance Formula

cuemath THE MATH EXPERT



$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

```
### Q1. Mean, var, std manually
arr = [2, 7, 3, 12, 9]
mean = (arr[0] + arr[1] + arr[2] + arr[3] + arr[4]) / len(arr)

variance = (((arr[0] - mean)**2) + ((arr[1] - mean)**2) +
            ((arr[2] - mean)**2) + ((arr[3] - mean)**2) +
            ((arr[4] - mean)**2)) / len(arr)
std = (variance)**0.5
print(mean, variance, std)

### 2
# Your solution looks good but to make it clear
# you can assign better variable names like below
p1 = [3, 2]
p2 = [4, 1]
x1 = p1[0]
y1 = p1[1]
# fill these question marks
x2 = p2[0]
y2 = p2[1]
ed = (((x2 - x1)**2) + ((y2 - y1)**2))**0.5
print(ed)


“””
# p1[0] = 3
This statement ( or value) means whatever is on the right-hand side
evaluate and assign that value to the variable on the left-hand side

x1 = 5
p1[0] = x1
p1 = [5, 2]

x1=p1[0]

X = 4
4 = x

x == 4
4 == x

= assignment operator: assign the value of right-hand side of the
variable/memory address or the left-hand side
```

```
== is called the ‘equal to’ operator it gives a boolean value
(True/False) after evaluating the expression on the left-hand side
and right-hand side
"""
```

```python
# Find mean, variance and standard deviation of [2, 7, 3, 12, 9] using for loop

# Given array
arr = [2, 7, 3, 12, 9]
# get the length of array (number of elements in array)
arr_len = len(arr)
# initialize variable arr_sum as 0 (0 is identity element to addition operation)
arr_sum = 0
# Looping through the array
for current_ele in arr:
   # adding current element to the array sum
   # Notice: arr_sum + current_ele will be evaluated first because it's on
   # the right hand side of expression
   # Then arr_sum variable will be overridden with newly calculated arr_sum + current_ele
   arr_sum = arr_sum + current_ele

# mean is sum of all observations / number of observations
mean = arr_sum / arr_len

variance_temp = 0
for current_ele in arr:
   variance_temp = variance_temp + ((current_ele - mean) ** 2)

# variance formula -> summation((current_element - mean) ^ 2) / n
# in the above we calculated only summation((current_element - mean) ^ 2) so
# / n is missing from variance
```

```
variance = variance_temp / arr_len

# standard deviation formula -> square_root(variance)
standard_deviation = variance ** 0.5

print(mean, variance, standard_deviation)
```