Absolutely! Let's break down the code line by line to understand it in detail:

**Defining the Data**

# Student Marks

student\_marks = [['Name', ['A','B','C','D','E']],

['Ankit', [41, 34, 45, 55, 63]],

['Aravind', [42, 23, 34, 44, 53]],

['Lakshay', [32, 23, 13, 54, 67]],

['Gyan', [23, 82, 23, 63, 34]],

['Pranav', [21, 23, 25, 56, 56]]]

* We define a list called student\_marks that contains the names of students and their marks in five subjects (A, B, C, D, E).
* The first element of the list is the header containing 'Name' and the list of subjects.
* The following elements are lists representing individual students and their corresponding marks in each subject.

**Function to Find the Student with the Highest Marks in a Particular Subject**

**# Function to find the student with the highest marks in a particular subject**

**def highest\_in\_subject(student\_marks, subject\_index):**

**highest\_marks = -1**

**top\_student = ""**

**for student in student\_marks[1:]:**

**name = student[0]**

**marks = student[1]**

**if marks[subject\_index] > highest\_marks:**

**highest\_marks = marks[subject\_index]**

**top\_student = name**

**return top\_student, highest\_marks**

**# Student Marks**

**student\_marks = [['Name', ['A','B','C','D','E']],**

**['Ankit', [41, 34, 45, 55, 63]],**

**['Aravind', [42, 23, 34, 44, 53]],**

**['Lakshay', [32, 23, 13, 54, 67]],**

**['Gyan', [23, 82, 23, 63, 34]],**

**['Pranav', [21, 23, 25, 56, 56]]]**

**# Function to find the student with the highest marks in a particular subject**

**def highest\_in\_subject(student\_marks, subject\_index):**

**highest\_marks = -1**

**top\_student = ""**

**for student in student\_marks[1:]:**

**name = student[0]**

**marks = student[1]**

**if marks[subject\_index] > highest\_marks:**

**highest\_marks = marks[subject\_index]**

**top\_student = name**

**return top\_student, highest\_marks**

**# Find the student with the highest marks in subject B**

**subject\_index = 1 # Index for subject B**

**top\_student, highest\_marks = highest\_in\_subject(student\_marks, subject\_index)**

**print(f"Top student in subject B: {top\_student} with marks {highest\_marks}")**

def highest\_in\_subject(student\_marks, subject\_index):

* This line defines a function named highest\_in\_subject which takes two parameters: student\_marks (the data) and subject\_index (the index of the subject we are interested in).

python

def highest\_in\_subject(student\_marks, subject\_index):

* This line defines a function named highest\_in\_subject which takes two parameters: student\_marks (the data) and subject\_index (the index of the subject we are interested in).

python

highest\_marks = -1

* We initialize a variable highest\_marks with a value of -1. This will store the highest marks found as we iterate through the data.

python

top\_student = ""

* We initialize a variable top\_student as an empty string. This will store the name of the student with the highest marks in the specified subject.

python

for student in student\_marks[1:]:

* We start a loop to iterate through each student in student\_marks, starting from the second element (index 1) to skip the header.

python

name = student[0]

marks = student[1]

* For each student, we extract the name (first element) and marks (second element, which is a list of marks).

python

if marks[subject\_index] > highest\_marks:

highest\_marks = marks[subject\_index]

top\_student = name

* We check if the marks in the specified subject (using subject\_index) are higher than highest\_marks.
* If so, we update highest\_marks with these marks and set top\_student to the name of the current student.

python

return top\_student, highest\_marks

* After the loop finishes, we return the top\_student and their highest\_marks in the specified subject.

**Using the Function to Find the Top Student in Subject B**

python

subject\_index = 1 # Index for subject B

* We set the subject\_index to 1, which corresponds to subject B in our data.

python

top\_student, highest\_marks = highest\_in\_subject(student\_marks, subject\_index)

* We call the highest\_in\_subject function with our data and the subject index for B.
* The function returns the top\_student and their highest\_marks in subject B.

python

print(f"Top student in subject B: {top\_student} with marks {highest\_marks}")

* We print out the name of the top student and their marks in subject B.

**Example Output**

When you run this code, the output will be:

Top student in subject B: Gyan with marks 82

If you have any more questions or need further clarification, feel free to ask