

6A)

1. `split(separator)`:

Use: `split()` splits a string at any point where the specified separator occurs. For instance, splitting with `,` splits at the comma.

In this program, `split(",")` is used to split the input string at every comma, turning it into a list of individual words.

2. `upper()`:

Use: `upper()` converts all lowercase letters in a string to uppercase.

We use the `upper()` function on every word to ensure that once we split the words, all of them will be in uppercase in this program.

3. `join(delimiter)`:

This serves for the use of taking a list of strings and combining them into one single string with each element being separated by the delimiter used. As in this program, `"".join(uppercase_words)` is used to join all words in the list `uppercase_words` with spaces between them to make a single sentence.

Modifications for Different Types of Separators

To get it to work for all kinds of separators, not just commas, we can replace the `split()` and `join()` methods:

- 1) **Splitting with Different Separators** : Instead of always calling `split(",")`, we can change it to a different separator, like `split("-")` for hyphens or `split("|")` for pipes. That way, if the person who inputs uses a different separator, we can still break the words correctly. Another choice is to ask the user what separator they used, then do it to `split()`. This makes the program more flexible since it is able to handle many formats.
- 2) **Combining with Other Separators**: The other thing that we can do using the `join()` method is to use some other character instead of a space. Let's say we would like our words joined with hyphens. Then we should use `"".join(".")` instead of `"".join(" ")`. Therefore, we will be able to choose the appearance of our final output.

Thus, by altering the behavior of `split()` and `join()`, this script can be applied to a wide range of input formats and can print output in different styles.

6B)

Documentation of String Preprocessing Methods

1. `replace(" ", "")`:

Purpose: The `replace()` method removes all spaces within the string by replacing each space with an empty string (`""`). This is helpful because we don't want spaces interfering with checking whether the word or phrase is the same forwards and backwards.

2. `lower()`:

Purpose: The `lower()` method will take all the uppercase letters in the string and change them to lower case.

This is the reason we do not want to use the capitalization while checking for a palindrome. For example, "Madam" must be treated just like "madam".

3. `strip()`:

Purpose: Removing spaces from both leading and trailing edges of an input.

This would be useful to prevent any extra space around the text from having an impact on our result.

Logic to Check Palindrome

Check the word or phrase is a palindrome by comparing the processed string with its reverse.

To reverse a string in Python use slicing: `[::-1]`.

Comparison: If the `processed_string` is the same as `processed_string[::-1]`, then it's a palindrome, because it reads the same forwards and backwards.

8A)

Function Documentation

`calculate_mean(grades)`:

Purpose: Calculates the mean value of grades

Parameters: (Grades) list of numeric values

Returns the average of grades as a float

Logic: The sum of all grades is added together then divided by `len(grades)` which is the number of grades.

`calculate_max(grades)`:

Purpose: Recovers the maximum value of grade from the list

Parameters (Grades): list of numeric values

Returns the maximum grade as a float

Logic The `max(grades)` function returns the greatest number in the list

`calculate_min(grades):`

Purpose: Get the lowest grade in the list.

grades: List of numerical grades.

Return: Lowest grade as a float.

Logic: `min(grades)` returns the lowest number in the list.

`calculate_std_dev(grades, mean):`

Purpose: Calculates the standard deviation of the grades, indicating how spread out the grades are from the mean.

grades: List of numerical grades.

mean: float, the average of the grades previously calculated.

Return: The standard deviation of the grades as a float.

Logic:

Calculating variance In this step we get the average of the squared differences of each grade from the mean.

Calculating standard deviation The square root of the variance is the standard deviation; so, `math.sqrt(variance)` returns the value.