Test-plan for bmi program (pytest)

Pytest is a testing framework used in Python that runs tests to make sure the code in a program is working as expected. This is often done by writing functions called tests, which contain checks that ensure the code runs as needed. For example, writing a test check on a function that provides the sum of two numbers. After a test check is written on a function, pytest then runs these test checks and reports back which ones passed and which ones failed. This allows programmers to spend much less time manually testing the input and output of code.

The reason I decided to go with pytest instead of other testing frameworks like unittest is because pytest seemed to be the simplest one to use, as it lets you easily test code by just writing **test_** in front of a function, making it much more user-friendly in my opinion. Another reason I chose pytest was because when it runs the tests in the command prompt, it provides a clear and simple report showing not only what went wrong in the test but also the value of the variable when it fails, making it very helpful for debugging purposes.

Function Tested:	Description:	Input:	Output:
test_convert_weight	Tests if the function correctly converts weight from pounds to kilograms. This is important to ensure accurate BMI calculation.	220 pounds	99.79024 kg
test_convert_weight_invalid	Tests that the function raises a ValueError when the weight is outside the allowed range. This prevents invalid inputs from continuing.	2000 pounds	ValueError raised
test_convert_height	Tests if the function correctly converts height in feet and inches to meters. This ensures accurate conversion for BMI calculations.	5 feet, 15 inches	ValueError raised

test_calculate_bmi	Tests if the function correctly calculates BMI. BMI must be accurate because it determines BMI ranges.	70 kilograms, 1.75 meters	22.857142857142858

Test case run