Q1

# Given variables

name = "Meiying"

lab1\_grade = 78

lab2\_grade = 90

midterm = 89

final = 30

# 1. Using % formatting

output1 = "Student name: %s\n- Lab grade:\n\tlab1\_grade: %d\n\tlab2\_grade: %d\n- midterm: %d\n- final: %d" % (name, lab1\_grade, lab2\_grade, midterm, final)

# 2. Using .format() method

output2 = "Student name: {}\n- Lab grade:\n\tlab1\_grade: {}\n\tlab2\_grade: {}\n- midterm: {}\n- final: {}".format(name, lab1\_grade, lab2\_grade, midterm, final)

# 3. Using f-strings (Recommended)

output3 = f"Student name: {name}\n- Lab grade:\n\tlab1\_grade: {lab1\_grade}\n\tlab2\_grade: {lab2\_grade}\n- midterm: {midterm}\n- final: {final}"

# 4. Using string concatenation (Less recommended)

output4 = "Student name: " + name + "\n- Lab grade:\n\tlab1\_grade: " + str(lab1\_grade) + "\n\tlab2\_grade: " + str(lab2\_grade) + "\n- midterm: " + str(midterm) + "\n- final: " + str(final)

Q3

def average\_rating(rating\_1: int, rating\_2: int, rating\_3: int) -> float: """ Return the average rating given that all ratings are between 1 and 5. """

Precondition: rating\_1 int

Precondition: rating\_2 int

Precondition: rating\_3 int

Q4

import doctest

def is\_palindrome(s: str) -> bool:

    """Return True if and only if s is a palindrome(e.g symmetric). For example, noon is a palindrome, but hello is not

    Precondition: s str

    >>> is\_palindrome("noon")

    True

    >>> is\_palindrome("hello")

    False

    >>> is\_palindrome("1")

    True

    >>> is\_palindrome("a")

    True

    >>> is\_palindrome(1)

    Traceback (most recent call last):

    ...

    AssertionError: invalid argument: s does not have the right type!

    >>> is\_palindrome(1.0)

    Traceback (most recent call last):

    ...

    AssertionError: invalid argument: s does not have the right type!

    >>> is\_palindrome(a)

    Traceback (most recent call last):

    ...

    NameError: name 'a' is not defined!

    """

    assert type(s) == str , "invalid argument: s does not have the right type!"

    return s == s[::-1]

doctest.testmod()

print(is\_palindrome("noon"))