Creating Simple Functions

Using appropriate function and variable names, write functions to accomplish each task. Call your function from the main program (multiple times, if indicated).

1. Given a name and a positive integer, display the name that many times.
2. Given a positive integer and a single character, display a "square" made of that character with a side length equal to that integer. For example, if the user enters 4 and the letter A, the square should appear as
   AAAA
   AAAA
   AAAA
   AAAA
3. Given an integer, 1-7, display the corresponding day of the week (e.g. 1 = Sunday). Display an "error" message for an invalid value (e.g. 9 = error).
4. Given a positive integer, calculate the sum of all *odd* integers up to (and possibly including) that integer.
5. Given two positive integers, m and n, determine and display the sum of the values from m to n (inclusive). Your function should handle all cases where $m < n$, $m > n$ or $m = n$.
6. *Bingo Draw*: Write a function that generates a letter-number combination (e.g. B12) representing a square on a traditional Bingo card. By convention, each letters has the following range:
   - B: 1-15
   - I: 16-30
   - N: 31-45
   - G: 46-60
   - O: 61-75

   Write a loop that calls your function 20 times. What is a potential issue with the program?
7. Write a function that takes the three sides of a triangle, and calculates the measures of all three angles. Output a suitable message if the triangle is impossible. Hint: is a triangle with side lengths of 1, 2 and 10 possible? How can you tell?
8. *Splash Zone*: A video game involves soaking enemies using water balloons. If a balloon hits the ground and explodes, water sprays outward in a circular pattern. If an enemy is very close to the balloon when it explodes, (s)he is in the *inner splash zone* and gets wetter than someone who is further from the explosion in the *outer splash zone*. Write a function that takes six arguments. The first two arguments specify the x- and y-coordinates of an object. The next two specify the x- and y-coordinates of the explosion. The final two specify the radii of the inner and outer splash zones. Based on the values given, determine whether the object falls inside of the inner splash zone, inside of the outer splash zone, or outside of both splash zones. You may find it useful to use the formula of a circle, centred at (p, q): $(x-p)^2+(y-q)^2=r^2$.