(d) **[2 marks]** Given the following code:

```
def my_function(n: int) -> None:
    """Precondition: n >= 2"""
    i = 4
    while i < n * n:
        i += 7
```

Find a formula for the *exact number of iterations* that this while loop runs, in terms of the input value n. You will need to use one of the floor or ceiling functions to ensure your expression is an integer. Show your work (do not just write a final expression).

The assignment '$i=4$' takes one step. After 'k' iterations of while loop, $i_k = 4+7k$ for this to be $\geq n^2$, $4+7k \geq n^2 \Rightarrow k \geq \frac{n^2-4}{7}$.
To make it an integer, the while loop stops after $\lceil \frac{n^2-4}{7} \rceil$ iterations and each iteration takes one step. Thus the total number of steps in this function are $1 + \lceil \frac{n^2-4}{7} \rceil (1) = 1 + \lceil \frac{n^2-4}{7} \rceil$.

The exact number of iteration that this while loop runs is $\lceil \frac{n^2-4}{7} \rceil$

(e) **[2 marks]** Complete the body of the function beloew so that its running time is $\Theta(n \log n)$. Only add code to the locations indicated by the boxes. Your code should use only variables, arithmetic and comparison operations, and (re)assignment statements. You may **not** call any functions (e.g., `math.log`).
No explanation is necessary; just write the code.

```
def f(n: int) -> None:
    """Precondition: n >= 0"""
    for _ in range(0, n):
        a = 1

        while  a < n  :        # Fill in the while loop condition

            a= a * 2           # Fill in the loop body (with one or more statements)
```