Problem 1

The idea is to generate a diamond shape whose width and height are equal to 5. The diamond must also be approximately, or exactly centered to the dimensions given. Note, this algorithm only works if height and width are equal or greater than 7. Height and width can be independent of each other. Also note that all division is assumed to be truncated. As in, anything after the last whole number is simply dropped.

To achieve this, there needs to be two loops, one nested in another. The first loop will go from the first row, all the way to the last; note that the first row will equal zero. The second loop will draw out the width of that row.

In the loops, there requires a conditional. The conditional will test to see how far from the center row the current row is from. To determine how far away from the center row, the simple equation can be used: $\left|\frac{h}{2} - i\right| = x$, where h is the height of the wall and i is the current row the loop is on; again i starts at zero. If $x \in \{0, 1, 2\}$ then the current row contains black tiles to place. Otherwise draw out white tiles only, which can be handled by a simple for loop which iterates over the width of the wall.

If there are black tiles to place, there requires one more logical statement. To determine how many tiles of the black color to place for a row, simply use: $2 \times \left|\left|\frac{h}{2} - i\right| - 2\right| + 1$. But to use this knowledge to effectively draw a diamond, first calculate: $\left|\left|\frac{h}{2} - i\right| - 2\right| = c$. c is the count of the tiles to the flanks of the middle tile. Using this value, then plug it in to a conditional in the row building for loop. The conditional will check to see if the current width position falls between: $\frac{w}{2} - c \leq pos \leq \frac{w}{2} + c$. If it does, it places a black tile instead of a white tile. Note when c is 0, it simply becomes true for the center only, meaning there are no flanks.

Below is a sample, python like pseudo code of the algorithm.

Listing 1: Code

```
# note in Python 3 // operator is truncated division.

for row in range(0,height):

    if (0 <= abs(height//2-row) <= 2):

        count = abs(abs(height//2-row)-2))

        for column in range(0,width):

            if (width//2-count <= column <= width//2-count)
                print("*")
            else
                print(" ")

    else

        for column in range(0,width)
            print(" ")
```