My Process to handle the task

1. As mentioned in the extra context the number of values must be flexible for testing purposes
   1. My solution to this was to wrap it in a function where the user can determine the width and height of the grid and by extension the number of elements
2. The next problem I had to tackle was generating a 2-D array with the dimensions specified in the previous stage and allow random values in a range to serve as the contents of the array
   1. I handled this using numpy to create a 2-D array with values between 1 - 100 and within the given dimensions
3. Next issue I faced was iterating through a 2-D array with increments of 3, I wasn’t sure how to do this so I decided to flatten the array and increment through it using a counter variable to check if 3 increments had passed and consider each element within that range an adjacent group of 3
   1. After 3 increments the value of the counter would be reset to count for the next group
4. Next, to determine what the highest available sum is, I decided that each group of 3 would be contained in a list and the sum of values in this list would be compared against the current greatest sum. If the new sum is greater than the current then it would be overwritten