

Carly Smith  
3414419

### Peer Problem: Shoot The Targets

I originally thought this problem was a lot simpler than it is. My original thought process was to find the minimum element in the array, use that as the target, and multiply it by its adjacent targets (taking boundaries into account), adding it to a running sum, and removing it from the array. Although this method worked for the last two test cases, I was only getting a maximum point score of 70 with the first test case input (as opposed to 167). I then tried a few separate trial and error target hits/multiplication with the first input test case to find how to achieve the maximum score. That's when I realized the algorithm needed for this problem. If the array was larger than two elements, I found the smallest element not including the boundaries and used it as a target. When the array got down to a size of 2, I included the boundaries, and when it was a size of one I just added the last element to the running sum. It was at this point that I encountered a problem with the last two input test cases and realized this wouldn't work if the boundaries were zero (as you would be multiplying the inner elements by zero and getting rid of them). So I realized I needed to add another check, to make sure I got rid of zero boundary elements first. This seemed to solve my problem, and is the final algorithm I am submitting.

I liked that this question had a math aspect to it - I enjoy figuring out number problems! I found it very straightforward to understand in terms of clarity, and thought the difficulty level was appropriate, as it was not as simple as finding the smallest value. As for improvements, I think the sample input could come in as just a list of single space separated numbers. Adding the square brackets and commas only adds extra coding time to read input and tests us on how well we know the Scanner class (useDelimiter to parse out the commas), not on our algorithmic skills.