Coding with reflection

Puzzle: **Temperatures**
Language: **Python 3**

After reading the instructions and starter code, my process began with brief strategizing in my head, then coding and ironing out the bugs along the way.

Without writing anything down, I decided I would use the absolute value function to compare the current closest with the next input. I used an array to keep the number closest to 0 at index 0. I used an elif block in case the absolute values were equal and stored t if it is positive.

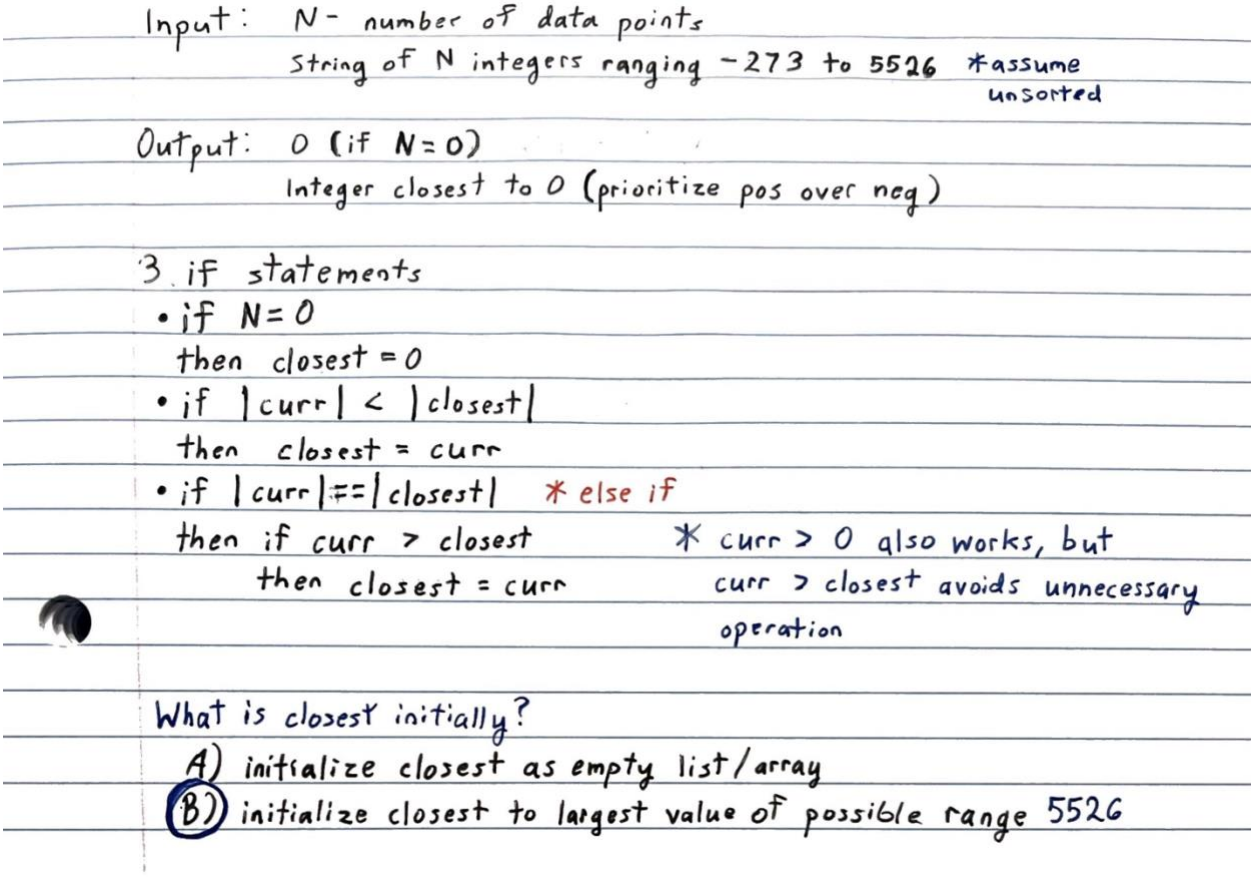
Upon reflecting, some things I noticed were that using a list was not necessary since I only needed the closest value and had no reason to store all the data points. I also realized some lines may be executed when unnecessary, such as storing t if it is positive even though closest may already be equal to t and positive. I also realized there was no point in taking the absolute value of the temperature minus zero.

Code before reflecting:

```
import sys
import math


# Auto-generated code below aims at helping you parse the standard input
# according to the problem statement.

n = int(input())  # the number of temperatures to analyse
closest = []
for i in input().split():
    # t: a temperature expressed as an integer ranging from -273 to 5526
    t = int(i)
    closest.append(t)
    if (abs(t-0) < abs(closest[0]-0)):
        closest[0] = t
    elif (abs(t-0) == abs(closest[0]-0)):
        if (t > 0):
            closest[0] = t

# Write an answer using print To debug: print("Debug messages...",
```

```
# file=sys.stderr, flush=True)
if (n==0):
    closest.append(0)


print(closest[0])
```

Although my initial solution passed every testcase, it was very sloppy. I started over by writing down the expected inputs and outputs. It was obvious that if statements were necessary, so I determined I would need about 3 if/elifs in total. I realized comparing the values of the current integer and the current closest was better than just saving the current temperature if it's positive. I thought about what closest should initially be, and I thought this might be the reason to use a list. However, since I was given the range of possible temperatures, I decided to use the value farthest from 0 to start.



By strategizing on paper rather than in my head, I felt more prepared and confident to solve the problem, and by using pseudocode in my plan, I also had a better idea of what functions to use and how to order my statements. By comparison, my code looked much cleaner and easier to follow than my initial attempt. I believe my final solution was also more efficient, both time- and memory-wise.