When dealing with addition the book says the presence of two operands with the same sign will result in a sign of the opposite. I started by creating a temporary result variable to store the result + increment so preconditions happen. In the first condition I am checking that result and increment are greater than zero if they are and the temp\_result is less than the numeric minimum than it will change the result code value to OVER and then return result. For the second condition I do the exact opposite, I check that the result and increment are less than zero and the temp\_result is greater than the numeric maximum. In the case of the third condition, I check if the temp\_result is less than result, this would indicate that a wraparound has taken place because the value has passed the maximum which results in the value being less than result. In this case I change result code to WRAP and return result.

For the subtraction conditions it is kind of the same, I ran into a major problem when trying to use the numeric minimum and maximums. The functions were not returning the correct information. I also tried following the book and the values returned were still not correct. So the solution included was the only viable solution I could come up with. As with the addition function the first two connections are almost the same with the difference being instead of using the numeric min and max I use zero which I change the result code to UNDER. Then on the third condition it is the exact opposite of the addition function.

This all comes together in the underflow and overflow functions. When each function is executed, and the result function is ran I use a simple if statement to check the result code. Based on the result code I then output a particular message and the result value.

The logic I used in this exercise comes directly from the book. For addition, if two integers are equal operands then the result will be opposite. For the wrap around condition, I used my own logic which was aided by the book. For subtraction, if the second operand is positive then check that the first operand is less than the minimum and if the second operand is negative then check that the first operand is greater than the maximum. Below is the screenshots of my results.

Text

Description automatically generated  
  
  
Text

Description automatically generated