Threat	Getting prize from the bank even when the conditions are not met.
Affected component	Banks money
Module details	Maman 12 Question 1 is_entitled_for_promotional_gift method line 2
Vulnerability class	Authorisation Inconsistency
Description	Since the limit **bound** is of type unsigned int and the **credit** is of type normal int, as soon as Yael's credit is in the minus she passes the test (**credit >= bound**) and she can receive a bonus even though she is in the minus (and not over 750). And this is because the test at the end is according to the unsigned method because of the conversion of **credit**. And when there is a negative number in the signed method (as **credit** was), the MSB has the number 1, which means that in the unsigned method, a **credit** that is negative is a very large number (much larger than 750), so at the end of the method the test is passed and at the end true is returned .
Result	Account holder gets money even if he doesn't have sufficient credit.
Prerequisites	The account holder needs credit less than zero.
Business impact	The bank loses money because of the vulnerability.
Proposed remediation	If you change the type of **bound** from unsigned int to int then the problem is solved because the two variables are already of the same type which is int and the test in the last line is based on the type of the two variables and therefore the method will return the required answer even if the credit is less than zero. unsigned int bound = 750; -> int bound = 750;