To prevent a buffer overflow in the provided code, it is important to ensure that the user cannot input more characters than the allocated space for the `user\_input` array. In the original code, the `user\_input` array is defined with 20 characters, but allowing the user to enter more than 19 characters would result in a buffer overflow and potentially overwrite the `account\_number` variable.

The solution to this problem is to limit the number of characters the user can input. By using `std::cin.width(19)`, the input is restricted to 19 characters, reserving the remaining space for the null terminator (`'\0'`). This ensures that the `user\_input` array does not overflow.

If the user enters more than the allowed number of characters, the code checks for this condition using `std::cin.fail()`. If the input is too long, the program clears the input error flag and discards the excess characters with `std::cin.ignore()`.

This approach prevents buffer overflow while preserving the integrity of the `account\_number` and provides appropriate feedback to the user when their input exceeds the allowed length.

A screenshot of a computer

Description automatically generated