

Candidate briefing

Hello!

Please find below a coding problem we'd like you to solve.

The purpose of this test is not to get a "right" or "wrong" answer - it's to see how you code. So, show us what you've got!

We'd normally expect you to spend an hour or so coding the solution.

You should submit your solution in a way that makes it easy for us to use. You should zip up the code, and include Visual Studio .sln files if using .Net; if you're using Java, we'd expect to see an Ant or Maven build script. Unless you include documentation to tell us otherwise, we will expect to be able to build and run the solution with no changes.

It should be quick and easy for us to verify that the input does indeed lead to the expected output.

What are we looking for?

We want to see you can create production-quality code. This means naming conventions, coding style, sensible design and meaningful commenting. If you feel you can give your work to a brand new colleague with a minimal of hand-over, you've probably got it right.

We are big fans of Test Driven Development, so including unit tests is a good idea.

We realize that writing even a trivial application, and having it "production ready" is a lot of work - so it's okay to leave "to do" comments in your code. Show us the intent, but don't write too much boilerplate.

Show that you're coding something that needs to be maintainable.

Let us know how long you spent on it and where you did it.

What are we not looking for?

We don't expect you to use every design pattern you've ever heard of - only apply patterns when it makes sense to do so.

We don't expect you to build a user interface - a command line application or unit test is fine.

We're not expecting you to have optimized the solution for performance or memory size - readability is more important than performance.

The whole thing built in the shortest amount of time - it's not a race. We'd like to see what you'd do under normal conditions - if you run out of time, just say so.

The Brief

You are tasked with developing software for an ATM machine. The software is responsible for validating customer account details and performing basic operations including balance inquiries and cash withdrawals.

A third party is developing the UI and will provide data to the application in an agreed format defined below. The application should receive the data, process the operations and then output the results in the required format also defined below. For the purposes of the test you are free to implement any mechanism for feeding input into your solution. You should provide sufficient evidence that your solution is complete by, as a minimum, indicating that it works correctly against the supplied test data.

The solution should meet the following business requirements:

- The ATM cannot dispense more money than it holds.
- The customer cannot withdraw more funds than they have access to.
- The ATM should not dispense funds if the pin is incorrect.
- The ATM should not expose the customer balance if the pin is incorrect.
- The ATM should only dispense the exact amounts requested.

Input

The first line is the total cash held in the ATM followed by a blank line. The remaining input represents zero or more user sessions. Each session consists of:

- The user's account number, correct PIN and the PIN they actually entered. These are separated by spaces.
- Then, on a new line, the customer's current balance and overdraft facility.
- Then, one or more transactions, each on a separate line. These can be one of the following types:
 - Balance inquiries, represented by the operation code B.
 - Cash withdrawals, represented by the operation code W followed by an amount.
 - A blank line marks the end of a user session.

Test Input
8000
12345678 1234 1234
500 100
B
W 100
87654321 4321 4321
100 0
W 10
87654321 4321 4321
0 0
W 10
B

- In the test data above the ATM is initialized with £8000 cash. The first customer has an actual account number 12345678 and pin number 1234. The customer entered the correct pin number 1234. The customer has a balance of £500 and overdraft facility of £100 (allowing them to withdraw £600 in total). The customer performed 2 operations, including a balance inquiry and a cash withdraw of £100.

Output

For each customer operation the solution should respond with the remaining customer balance, or ACCOUNT_ERR if the account details could not be validated. If funds aren't available for cash withdraw the required response is FUNDS_ERR. If the ATM is out of cash the required response is ATM_ERR.

The response to the test data above would be:

Test Ouput
500
400
90
FUNDS_ERR
0