

## BANK ACCOUNT - June 2023 Paper 22

A two-dimensional (2D) array `Account[]` contains account holders' names and passwords for a banking program.

A 2D array `AccDetails[]` has three columns containing the following details:

- column one stores the balance – the amount of money in the account, for example 250.00
- column two stores the overdraft limit – the maximum total amount an account holder can borrow from the bank after the account balance reaches 0.00, for example 100.00
- column three stores the withdrawal limit – the amount of money that can be withdrawn at one time, for example 200.00

The amount of money in a bank account can be negative (overdrawn) but **not** by more than the overdraft limit.

For example, an account with an overdraft limit of 100.00 must have a balance that is greater than or equal to  $-100.00$

Suitable error messages must be displayed if a withdrawal cannot take place, for example if the overdraft limit or the size of withdrawal is exceeded.

The bank account ID gives the index of each account holder's data held in the two arrays.

For example, account ID 20's details would be held in:

`Account[20,1]` and `Account[20,2]`

`AccDetails[20,1]` `AccDetails[20,2]` and `AccDetails[20,3]`

The variable `Size` contains the number of accounts.

*AccID starts at 1, ends at Size*

The arrays and variable `Size` have already been set up and the data stored.

Write a program that meets the following requirements:

- ✓ checks the account ID exists and the name and password entered by the account holder match the name and password stored in `Account[]` before any action can take place
- displays a menu showing the four actions available for the account holder to choose from:
  1. display balance
  2. withdraw money
  3. deposit money
  4. exit
- allows an action to be chosen and completed. Each action is completed by a procedure with a parameter of the account ID.

*CarryOn ← FALSE*

You must use pseudocode or program code **and** add comments to explain how your code works. All inputs and outputs must contain suitable messages.

You only need to declare any local arrays and local variables that you use.

You do **not** need to declare and initialise the data in the global arrays `Account[]` and `AccDetails[]` and the variable `Size`

```

REPEAT // Check valid Account ID number
    OUTPUT "please enter account ID"
    INPUT AccID
UNTIL AccID > 0 AND AccID ≤ Size
    CarryOn ← TRUE
    OUTPUT "Please enter your name" & password
    INPUT Name, Password
    OUTPUT "Please enter your password"
    INPUT Password
    IF Name <> Account[AccID, 1] THEN
        OUTPUT "Invalid name"
        CarryOn ← FALSE
    ENDIF
    IF Password <> Account[AccID, 2] THEN
        OUTPUT "Invalid password"
        CarryOn ← FALSE
    ENDIF
    WHILE CarryOn = TRUE DO // Loop to display options
        OUTPUT "Please enter one of the following..."
        OUTPUT "1... 2... 3... 4..."
        INPUT Choice
        CASE OF Choice
            1: Balance(AccID) // Proceed to...
            2: Withdraw(AccID) // Proc...
            3: Deposit(AccID) // Proc...
            4: CarryOn ← FALSE
        OTHERWISE OUTPUT "Invalid choice"
        ENDCASE
    ENDWHILE
ENDWHILE

```