Lab5 Report

崔士强 PB22151743

December 24, 2023

1 Program Design

The structure of the program is shown in the following figure. The function TERM stands for TERMination.

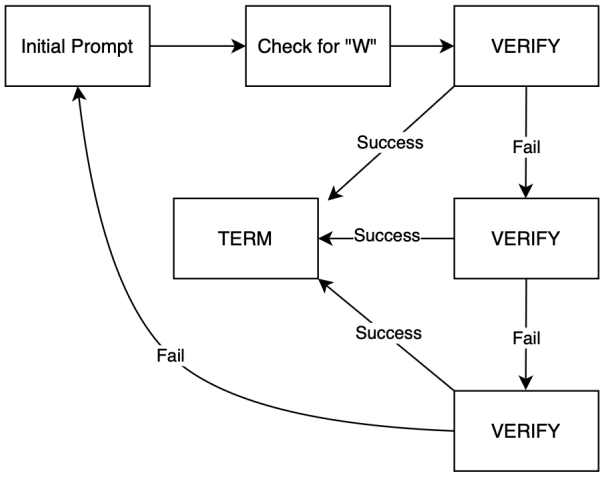

Figure 1: Overall structure

Here is the details of the function VERIFY. It reads a character each time and compare it to the corresponding character in the password. If an unmatching character is found or the length of input exceeds the password. The indicator(i.e. `R0`) is set negative.

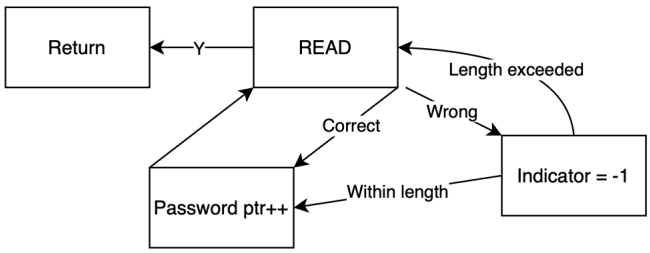

Figure 2: Details of VERIFY

2 Testing Evidence

Screen recording: Program test

3 Discussion Questions

3.1 Functions

Several functions are used in the program. Reasons:

1. Some part of the program may be executed more than once.
2. The use of functions increases readability and makes debugging easier.

3.2 Recursion

Recursive functions can be used in the program for checking password. But due to the simplicity of the process (10-character password and 3 attempts), I chose not to use them.

Also, the use of recursion to check password may have a problem when input is super long.

Actually attempts and preparations to use recursion was made.

3.3 Preset Prompts

In this program, only 3 attempts are allowed, which means prompts are not too many. In order to keep developing process simple, all prompts are stored using `.STRINGZ` , rather than combine prompts for incorrect inputs, which is a resonable approach if number of attempts allowed may be modified.

```
1  INITPRPT    .STRINGZ  "Welcome to the bank system! Type 'W' to withdraw some fund."
2  SUCCPRPT    .STRINGZ  "Success!"
3  FAILIPRPT   .STRINGZ  "Incorrect password! 2 attempts remaining."
4  FAILIIPRPT  .STRINGZ  "Incorrect password! 1 attempt remaining."
```

```
5   FAILIIIPRPT .STRINGZ  "Fails."
6   INPUTPRPT   .STRINGZ  "Please input your password:"
7   PSWD        .STRINGZ  "PB22151743"
```

3.4 Program Security

There are two kinds of inputs worth considering: empty input and long input.

In this program, the length of input won't cause a problem. Every time a character is read by the program, it's immediately checked. If the input is longer than the password or a wrong character is found, the indicator will be kept negative.

As for the empty input, A register is set to mark the first character. If the first character is Y, the indicator will be nagative.

3.5 Challenges

An error occured when I tried to assemble the program, which I had never met before:

Figure 3: An error

The same instruction worked well in Lab4, which baffled me. Later I realized what the message means: PCoffset was too large.

Solution: move the instruction up.