

Exercise 1: Change NUSNET Password

[50 marks]

Problem Statement

It is a university policy that all students and staff need to change their NUSNET passwords every half a year through the following graphic user interface (GUI):

Change NUSNET Password - Windows Internet Explorer
https://exchange.nus.edu.sg/iisadmpwd/

Change NUSNET Password

For NUS Staff, please key in **NUSSTF** under "Domain" and your userID under "UserID".

For NUS Students and Graduates, please key in **NUSSTU** under "Domain" and your userID under "UserID".

For NUS Alumni, please login to **AlumNet** to change your password.

For NUS Visitors, please key in **NUSEXT** under "Domain" and your userID under "UserID".

For NUHS Staff, please leave "Domain" blank and key in **firstname_lastname@nuhs.edu.sg** under "UserID".

Note:

- Your password must be at least 8 characters in length.
- Your password will expire in 180 days.
- Your password must be complex. Take the [password complexity test](#).
- Your password cannot contain your userID or any part of your name.
- You cannot re-use any of your 6 old passwords.
- You cannot change your password more than once in a day.

Domain:

UserID:

Old password:

New password:

Confirm new password:

In this exercise, you are to check whether a user has entered an acceptable new password. To keep things simple, we just check if the following rules are satisfied:

- The new password must be at least 8 characters in length.
- The new password must be complex: each character in a password can only be either a letter ('a'-'z', 'A'-'Z'), underscore ('_') or a digit (0-9). However, a complex password must contain both upper case and lower-case letters.
- The new password must not contain the UserID.
- The confirmation for the new password must match the new password.
- The new password must not be the same as the old password.

Write a program to read in sequence, *domain*, *UserID*, *old password* and *new password* (twice), check whether the new password is acceptable or not. It prints out **"Password changed successfully!"** if so, or **"Error: Invalid userID or password!"** otherwise.

Note that both UserID and passwords are case sensitive. You do not need to check the validity of old password. You may assume that no input data contains whitespace and each input is at most 15 characters long.

Write on the skeleton file **password.c** given to you. The **main** function is complete and you are to fill in the **isValidPassword** function only. You may write other helper functions if needed.

Check sample runs (next page) for input and output format.

Sample Runs

Six sample runs are shown below with user input highlighted in **bold**.

```
Domain: NUSSTF
UserID: DCSZHOUL
Old password: WillNotTellU
New password: CS1010_12s1
Confirm new password: CS1010_12s1
Password changed successfully!
```

```
Domain: NUSSTF
UserID: DCSZHOUL
Old password: TellMeLah
New password: DCSzhoUl
Confirm new password: DCSzhoUl
Password changed successfully!
```

```
Domain: NUSSTF
UserID: DCSZHOUL
Old password: Oh_MyGod
New password: CS1010_11s2
Confirm new password: CS1010_11s2
Error: Invalid userID or password!
```

No lower-case letter

```
Domain: NUSSTU
UserID: a0123456
Old password: A0123456x
New password: a0123456X
Confirm new password: a0123456X
Error: Invalid userID or password!
```

Contains userID

```
Domain: NUSSTU
UserID: a0123456
Old password: YesMadam
New password: Whatever
Confirm new password: WhatEVER
Error: Invalid userID or password!
```

Confirmation does not match new password

```
Domain: NUSSTU
UserID: a0123456
Old password: SoCstudent
New password: Zhou@comp
Confirm new password: Zhou@comp
Error: Invalid userID or password!
```

Contains illegal character '@'