

**NATIONAL UNIVERSITY OF SINGAPORE  
SCHOOL OF COMPUTING**

Practical Examination 2 (PE2) for Semester 1, AY2012/13  
**CS1010 — Programming Methodology**

27 October 2012

Time Allowed: 2 hours

---

**INSTRUCTION TO CANDIDATES**

1. You are only allowed to read this cover page. Do **not** read the question paper until you are told to do so.
2. This paper consists of **2** exercises on **6** pages. Each exercise constitutes 50%.
3. This is an open-book exam. You may bring in any printed material, but **not** electronic devices, including but not limited to laptop, thumb-drive, electronic dictionary and calculator. You are to switch off/silence your mobile phone and keep it out of view.
4. You will be logged into a special account at the beginning of the PE. Do not log off until the end of the PE. Do not use your own NUSNET account.
5. A plab account slip will be issued to you at the beginning of the PE. Bring your matriculation card for identification when you collect it. Please leave your matriculation card on the desk in front of you throughout the PE.
6. You are to write your program in the given **plab account**. The host name is **plab2** (not sunfire!). No activity should be done outside this plab account.
7. You **do not** need to submit your programs to CodeCrunch. We will retrieve your programs and submit them to CodeCrunch after the PE.
8. Skeleton programs are already residing in your plab account. Please leave the programs in the home directory, and use the same program names as specified in the paper. Do **not** create subdirectory to put your programs there or we will not be able to find them!
9. **Only your source codes (.c programs)** from your plab account will be collected after the PE. Hence, how you name your executable files is not important.
10. Please read carefully and follow all instructions in the question. If in doubt, please ask. Raise your hand and the invigilator will attend to you.
11. Any form of communication with other students, or the use of unauthorised materials is considered cheating and you are liable to disciplinary action.
12. Please save your programs regularly during the PE.
13. When you are told to stop, please do so immediately, or you will be penalised.
14. At the end of the PE, please **log out from your plab account** and **shut down the PC**.
15. Please check and keep your belongings (esp. matriculation card) before you leave.
16. We will make arrangement for you to retrieve your programs after we have finished grading. Grading may take a week or more.

---

**ALL THE BEST!**

---

## **CS1010 AY2012/13 Semester 1**

### **Practical Exam 2 (PE2)**

#### **Advice on Writing Your Programs – Please read!**

- You are advised to spend some time thinking over the tasks to design your algorithms, instead of writing the programs right away.
- If you write a function, you must have a function prototype, and you must put the function definition after the main function.
- You may write additional function(s) not mentioned in the question, if you think it is necessary.
- Any variable you use must be declared in some function. You are not allowed to use global variables (variables that are declared outside all the functions).
- You may assume that all inputs are valid, that is, you do not need to perform input validity check.
- Manage your time well! Do not spend excessive time on any exercise.
- The rough marking scheme for both exercises is given below.

#### **Rough Marking Scheme For Each Exercise [50 marks]**

1. Style: 10 marks
  - Top of program: Are name, matriculation number, plab account-id, DG and description filled?
  - Is there a description for every function except the main function?
  - Proper indentation and naming of variables?
  - Appropriate comments wherever necessary?
2. Design: 10 marks
  - Correct definition and use of functions?
  - Presence of function prototypes?
  - Is algorithm simple or unnecessarily complicated?
3. Correctness: 30 marks
  - The graders will go through your program and award marks manually.
4. Extra deductions (not restricted to the following):
  - Program cannot be compiled: Deduct 10 marks
  - Compiler issues warning with -Wall: Deduct 5 marks.
  - Use of global variables: Deduct 10 marks

## 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:

OK Cancel Reset

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 '@'

## Exercise 2: Poker Game

[50 marks]

### Problem Statement

A game of poker is played between two players. Each player is dealt a hand of five cards. The player with the higher-ranking hand wins the game.

In this exercise, you are to write a program to determine which player wins the game or there is a draw, given the hands of both players. To simplify the problem, we do not consider the suit of the cards.

A hand of cards is described by its **type** and **key card** as defined below:

| Type            | Rank | Definition                                | Key card                                  | Examples                                |
|-----------------|------|---|---|---|
| Four-of-a-kind  | 4    | Four of the cards are identical.          | One of the four identical cards.          | <b>2</b> 2 2 2 J<br>K <b>Q</b> Q Q Q    |
| Three-of-a-kind | 3    | Three of the cards are identical.         | One of the three identical cards.         | <b>3</b> 3 3 A A<br>8 8 <b>10</b> 10 10 |
| Pair            | 2    | One or more pairs of cards are identical. | The highest-ranking card among the pairs. | <b>9</b> 9 A 2 3<br>9 9 <b>J</b> J 5    |
| No-pair         | 1    | None of the cards are identical.          | The highest-ranking card in the hand.     | 5 <b>J</b> 2 4 8<br><b>Q</b> 9 7 3 6    |

(Note: the key card of each hand is highlighted in **bold**.)

A hand of a higher rank in type always wins another hand of a lower rank in type, regardless of their key card. For example, four-of-a-kind always wins three-of-a-kind, while pair always wins no-pair.

When two hands are of the same rank in type, the one with a key card of a higher rank wins. The rank of cards is as listed below.

| Card | Rank | Card | Rank | Card | Rank | Card | Rank |
|------|------|------|------|------|------|------|------|
| A    | 13   | 10   | 9    | 6    | 5    | 2    | 1    |
| K    | 12   | 9    | 8    | 5    | 4    |      |      |
| Q    | 11   | 8    | 7    | 4    | 3    |      |      |
| J    | 10   | 7    | 6    | 3    | 2    |      |      |

Your program should read in two arrays of five integers that represent the hands of the two players. The integers are between 1 and 13, corresponding to A, 2, 3, ..., J, Q, K, respectively. Your program should print messages indicating (1) the type ranks and the key card ranks of both hands, and (2) which player wins or there is a draw.

You may assume that the inputs are valid (*i.e.*, the integers are within the specified range and there would be no more than four cards of the same card number).

Write on the skeleton file **poker.c** given to you. You need to include two functions:

- **type\_and\_keycard\_ranks()** which takes in an array representing a hand, and returns both the rank of its type and the rank of its key card.
- **who\_wins()** which takes in the type rank and the key card rank of both hands, and returns 1 if player 1 wins, 2 if player 2 wins, or 0 if there is a draw.

You may define additional functions as needed. Check sample runs below for input and output format. The input and output files for these sample runs are available in your plab account.

## **Sample Runs**

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

```
Enter the hand of player 1:
2 2 2 2 11
Enter the hand of player 2:
3 3 3 1 1
Player 1 has a hand of type rank 4 and key card rank 1.
Player 2 has a hand of type rank 3 and key card rank 2.
Player 1 wins.
```

```
Enter the hand of player 1:
5 11 2 4 8
Enter the hand of player 2:
9 9 1 2 3
Player 1 has a hand of type rank 1 and key card rank 10.
Player 2 has a hand of type rank 2 and key card rank 8.
Player 2 wins.
```

```
Enter the hand of player 1:
2 2 2 2 11
Enter the hand of player 2:
13 12 12 12 12
Player 1 has a hand of type rank 4 and key card rank 1.
Player 2 has a hand of type rank 4 and key card rank 11.
Player 2 wins.
```

```
Enter the hand of player 1:
5 11 2 4 8
Enter the hand of player 2:
8 5 11 10 7
Player 1 has a hand of type rank 1 and key card rank 10.
Player 2 has a hand of type rank 1 and key card rank 10.
There is a draw.
```

```
Enter the hand of player 1:
9 9 11 11 5
Enter the hand of player 2:
10 10 1 2 3
Player 1 has a hand of type rank 2 and key card rank 10.
Player 2 has a hand of type rank 2 and key card rank 9.
Player 1 wins.
```