NATIONAL UNIVERSITY OF SINGAPORE
SCHOOL OF COMPUTING

Practical Examination 1 (PE1) for Semester 1, AY2010/1
CS1010 — Programming Methodology

18 September 2010 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 **7** pages.
3. This is an open-book exam. You may bring in any printed material, but **not** electronic devices, including thumb-drive, electronic dictionary and calculator. (You may use the Calculator in the computer's "Accessories".) You are to switch off/silence your mobile phone and keep it out of view.
4. You are to use your given **plab account** to write your programs. The host name is **plab2** (not sunfire). Bring your matriculation card along when you collect your plab account slip.
5. CourseMarker is **not** used in this PE.
6. Skeleton programs are already loaded into your plab account. Please code in the plab account's home directory. Do **not** create subdirectory to put your programs there, or we will not be able to find them!
7. **Only the source codes (.c programs)** from your plab account will be collected after the PE. Hence, how you name your executable files is not important.
8. Unless otherwise stated, you do not need to check the validity of input data.
9. Please put your matriculation card on the desk in front of you.
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. You are **not** allowed to use any software (apart from SSH Secured Shell to log into your plab account, and the Calculator in the Accessories) on your PC.
12. Any form of communication with other students, or the use of unauthorised materials is considered cheating and you are liable to disciplinary action.
13. Please save your programs regularly during the PE.
14. When you are told to stop, please do so **immediately**, or you will be penalised.
15. At the end of the PE, please **log out from your plab account** and **shut down the PC**.
16. Please check your belongings before you leave.

ALL THE BEST!

CS1010 AY2010/1 Semester 1
Practical Exam 1 (PE1)

Important notes

- There are **2** exercises in this PE, each constituting 50%.
- The marking schemes for both exercises are given below.
- You are advised to spend some time thinking over the tasks to design your algorithms, instead of writing the programs right away.
- You are **not** allowed to use pointers, arrays, recursion or any string functions. (Violation will result in zero mark for correctness.)
- If you write a function, you must have a function prototype, and you must put the function definition after the main function.
- 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).
- Manage your time well! Do not spend excessive time on any exercise.

Marking Scheme

Exercise 1: code.c [50 marks]

1. Can the program be compiled? 2 marks
2. Issues on function: 5 marks
3. Style: 10 marks
4. Issues on design: 3 marks
5. Correctness: 30 marks
    - Based on outputs (10 test data sets @ 3 marks each)
6. Use of pointers, arrays, recursion, string functions or global variables
    - No mark for correctness

Exercise 2: analyser.c [50 marks]

1. Can the program be compiled? 2 marks
2. Issues on function: 5 marks
3. Style: 10 marks
4. Issues on design: 3 marks
5. Correctness: 30 marks
    - Menu: 2 marks
    - Task 1 (minimum and maximum): 4 marks
    - Task 2 (average): 4 marks
    - Task 3 (frequency table): 6 marks
    - Task 4 (histogram): 14 marks
6. Use of pointers, arrays, recursion, string functions or global variables
    - No mark for correctness

Exercise 1: Game Code (50 marks)

A game software requires its player to enter the number of days, hours and minutes (all of type **int**) since the last time the player played the game. It then displays a 2-character game code according to the following steps:

Step 1: Convert the number of days, hours and minutes into minutes (you may assume that the value converted is a positive integer and can be represented in **int** type). For instance, if the user enters 2 3 15, the converted value is **3075**.

Step 2: Pick the 4^{th} and 1^{st} digits from the right of the value obtained in step 1, and put them together as a 2-digit number (in the numerical range 0 through 99). For instance, if step 1 yields 3075, then this step yields **35**. (If step 1 yields a value less than 1000, then the 4^{th} digit is 0.)

Step 3: Use the table below to generate the first character of the game code:

If the result of step 2…	***Then the first character of the game code is***
is divisible by 2	A
is divisible by 3	F
is divisible by 5	K
is divisible by 7	P
is divisible by 11 or by 13	T
is none of the above	Z

The table is checked from top to bottom. In our example, 35 is divible by 5 as well as 7. Hence the first character of the game code is **K**, not P.

Step 4: Take the square root of the result in step 2, and pick the two digits after the decimal point. Treat these two digits as an integer in the numerical range 0 through 99. For instance, step 2 yields 35, its square root is 5.916080, hence the result of this step is **91**.

Step 5: Using the result in step 4, follow the table below to generate the second character of the game code:

If the result of step 4…	***Then the second character of the game code is***
is divisible by 2	A
is divisible by 3	F
is divisible by 5	K
is divisible by 7	P
is divisible by 11 or by 13	T
is none of the above	Z

For our example of 91, the second character of the game code is **P**.

Step 6: Hence, the game code for our example is **KP**.

Write a program **code.c** to read in 3 values (using **int** type) representing the number of days, hours and minutes the user last played the game, and display the **game code**.

Skeleton Program:

A skeleton program **code.c** is available in your plab account and is shown below.

```
// CS1010 AY2010/1 Semester 1
// PE1 Ex1: code.c
// Name:
// Matriculation number:
// Discussion group:
// Description:

int main(void)
{
    int day_input, hr_input, min_input; // user's inputs

    printf("Enter days, hours and minutes: ");


}
```

Sample Runs:

Sample runs are shown below. User input is shown in **bold**.

```
Enter days, hours and minutes: 2 3 15
KP
```

```
Enter days, hours and minutes: 1 2 3
TA
```

```
Enter days, hours and minutes: 9 0 7
FZ
```

Exercise 2: Test Results (50 marks)

You are a lecturer for a class of students and you have just finished marking a test conducted for them. You want to analyse the results to see how your students performed. Write a C program called **analyser.c** to read in the set of marks scored for the test (which are integers ranging from 0 to 100 inclusively). An input of a negative number indicates the end of data input and is not to be included as part of the input data. It is assumed that the class consists of at least one student. The program then presents the user with the following menu of choices.

1. Show the maximum and minimum marks
2. Show the average mark
3. Show the grade frequency table
4. Show the grade histogram
5. Exit

(Please see the sample run below for the required output format.)

The grade assigned for a particular mark is based on the following table:

Mark	grade
Mark ≥ 80	A
70 ≤ Mark ≤ 79	B
60 ≤ Mark ≤ 69	C
50 ≤ Mark ≤ 59	D
Mark < 50	F

Notes:

1. You must use **float** type for the average mark, and display the average mark to 2 decimal places.
2. You are to use a **`switch`** statement to handle the choice selection of the menu. (If you use `if` statement instead, only partial credit will be given.)
3. You are to use a **`switch`** statement for the calculation of the grade frequency. (If you use `if` statement instead, only partial credit will be given.)
4. You may assume that no frequency is higher than 9, i.e. no more than 9 students can score a particular grade.
5. You must have a function to print the grade frequency table and a function to print the histogram.
6. You are free to write any additional function(s) you find necessary.

The **ex2_printf** file:

This file is in your plab account. It contains some `printf` statements you might want to use in your program. To insert this file into your **analyser.c** program, first **vim analyser.c**, go to the line where you want to insert, and (in command mode) type

```
:r ex2_printf
```

Skeleton Program:

A skeleton program **analyser.c** is available in your plab account and is shown below.

```
// CS1010 AY2010/1 Semester 1
// PE1 Ex2: analyser.c
// Name:
// Matriculation number:
// Discussion group:
// Description:

int main(void)
{
    int mark, choice; // user's inputs

    printf("Enter mark (negative number to end): ");

}
```

Sample run:

A sample run of the program is shown below. User input is shown in **bold**.

```
Enter mark (negative number to end): 40
Enter mark (negative number to end): 59
Enter mark (negative number to end): 70
Enter mark (negative number to end): 82
Enter mark (negative number to end): 61
Enter mark (negative number to end): 65
Enter mark (negative number to end): 75
Enter mark (negative number to end): 76
Enter mark (negative number to end): 55
Enter mark (negative number to end): 66
Enter mark (negative number to end): -10
1. Show maximum and minimum marks
2. Show average mark
3. Show grade frequency table
4. Show grade histogram
5. Exit
Select your choice: 1

The maximum and minimum are 82 and 40 respectively.

1. Show maximum and minimum marks
2. Show average mark
3. Show grade frequency table
4. Show grade histogram
5. Exit
Select your choice: 2

The average mark is 64.90.

1. Show maximum and minimum marks
```

```
2. Show average mark
3. Show grade frequency table
4. Show grade histogram
5. Exit
Select your choice: 3

Grade Frequency Table
=====================
A: 1
B: 3
C: 3
D: 2
F: 1

1. Show maximum and minimum marks
2. Show average mark
3. Show grade frequency table
4. Show grade histogram
5. Exit
Select your choice: 4

Grade Histogram
===============
3              ***  ***
2        ***  ***  ***
1   ***  ***  ***  ***  ***
     F    D    C    B    A

1. Show maximum and minimum marks
2. Show average mark
3. Show grade frequency table
4. Show grade histogram
5. Exit
Select your choice: 5

Thank you and goodbye!
```

IMPORTANT REMINDERS:

- Make sure your programs are called **code.c** and **analyser.c**, as only these two files will be picked up from your plab account. No other files will be picked up.
- Make sure your **code.c** and **analyser.c** are in the home directory of your plab account, and not hidden in some subdirectory, otherwise they will not be picked up.

=== END OF PAPER ===

=== HAVE AN ENJOYABLE RECESS! ===