NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING

EXAMINATION FOR
Semester 1 AY2011/2012

CS1010E — PROGRAMMING METHODOLOGY

Nov / Dec 2011 Time Allowed: 2 Hours

INSTRUCTIONS TO CANDIDATES

1. This examination paper consists of **FOURTEEN (14)** questions and comprises **SEVENTEEN(17)** printed pages, including this page.
2. Answer **ALL** questions.
3. Answer Section A (Questions 1 to 10) by shading the letter corresponding to the most appropriate answer on the OCR form provided.
4. Answer Section B (Questions 11 to 14) within the space provided in this booklet. You may use pen or pencil to write your answers.
5. This is an **OPEN BOOK** exam. The maximum mark is **80**.
6. Calculators are allowed, but not electronic dictionaries, laptops, PDAs, or other computing devices.
7. Do not look at the questions until you are told to do so.
8. Please write your **Matriculation number** below.

This portion is for examiner's use only.

Question	Marks	Remarks
Q11		
Q12		
Q13		
Q14		
Total		

SECTION B (4 Questions : 60 Marks)
Write your answers in the space provided.

11. (a) **[5 marks]** Complete the **recursive** function `sumPairs` that calculates the sum of all pairs of digits of a positive integer starting from the right to the left. For example, if the number is 2345678, the answer is $78 + 56 + 34 + 2$ which gives 170.

ANSWER:

```
int sumPairs(int num)
{
```

(b) **[5 marks]** Let x and y be two integers which may not have the same number of digits. Complete the **recursive** function `multiplyPairs` to multiply each of the corresponding digits of x and y, scanning from right to left. The products should be added up. If there are no more digits in the shorter integer, add the remaining digits of the longer integer to the sum. For example, if $x = 1234$ and $y = 123457$, the function should return 66, which is derived as follows:

$$4 \times 7 + 3 \times 5 + 2 \times 4 + 1 \times 3 + 12 = 28 + 15 + 8 + 3 + 12 = 66$$

ANSWER:

```
int multiplyPairs(int n1, int n2)
{
```

12. Given a matrix $\mathbf{A}$, the transpose of $\mathbf{A}$ is the matrix $\mathbf{A}^T$ whose rows are the columns of $\mathbf{A}$. For example,

$$\mathbf{A} = \begin{bmatrix} 2 & 4 & 5 \\ 1 & 8 & 2 \\ 9 & 8 & 0 \end{bmatrix} \qquad \mathbf{A}^T = \begin{bmatrix} 2 & 1 & 9 \\ 4 & 8 & 8 \\ 5 & 2 & 0 \end{bmatrix}$$

(a) [**5 marks**] Complete the function `transpose` that takes in square matrix `mat` of dimensions `n` × `n` and transposes the matrix.

ANSWER:

```
void transpose(int mat[][MAX], int n)
{
```

(b) [**5 marks**] A matrix is symmetric if it is equal to its transpose. Complete the function `isSymmetric` that takes in a square matrix `mat` of dimensions `n` × `n` and returns `TRUE` if the matrix is symmetric and `FALSE` otherwise.

ANSWER:

```
#define TRUE 1
#define FALSE 0
int isSymmetric(int mat[][MAX], int n)
{
```

13. We have seen in the lectures that a palindrome is a sentence or word that reads the same backwards or forwards. We define a sentence palindrome as one that reads the same when the order of the words of a sentence is reversed. For example,

```
I DO, do i?
```

is a sentence palindrome. Assume that the read sentence consists of only upper and lower case letters, punctuations, and blanks. You are only allowed to use the following string functions: `strcpy`, `strcat`, `strcmp` and `strlen`.

(a) [**7 marks**] Complete the function `trim` that takes a string `str` and modifies the string such that:

- Upper case letters are converted to lower case;
- Punctuations are removed;
- Consecutive spaces are trimmed to one space.

ANSWER:

```
void trim(char *str)
{
```

(b) [**8 marks**] Complete the function `isPalindrome` that takes in a sentence as a string `str` and returns `TRUE` if `str` is a sentence palindrome, and `FALSE` otherwise. You may only assume that there is at least one word in the sentence, but not the maximum number of words or maximum length of the sentence. You are allowed to modify `str`.

ANSWER:

```
#define TRUE  1
#define FALSE 0
int isPalindrome(char *str)
{
```

14. You are to develop a program to simulate the functions of a jackpot machine in a casino. This machine has 3 reels which will each generate randomly different objects and numbers, comprising `Flower`, `Cherries`, `Lion`, `Monkey`, and numbers `1`, `3`, `5`, and `7`. A gambler places a bet consisting of a number of coins, and the reels are spun. The gambler receives a payout if all three reels display the same object or number. The payout that the gambler will get is calculated by the number of coins bet raised to the power of the value as determined by the table below. The top prize is when all three reels show 7.

Object	Value
Flower	1
Cherries	2
Lion	2
Monkey	3
1	1
3	2
5	2
7	5

As an example, if the bet is 4 coins, and the reels show three `Lion` objects, then the payout is $4^2 = 16$.

You are to complete the following task within the program provided. You are advised to read through all the parts before proceeding to give your answer.

(a) **[3 marks]** Define a structure `struct Object` consisting of two members: the name of the object (`name`), and the associated value (`value`).

(b) **[3 marks]** In the `main` function, declare an array `reel` of type `struct Object` to hold all the eight different objects.

(c) **[1 mark]** In the `main` function, declare an array `roll` of type `struct Object *`. This will be used to store the outcome of a particular roll of the jackpot.

(d) **[3 marks]** Write a function `rolling` that takes in the arrays `reel` and `roll` and simulates one roll of the jackpot. Upon termination of the function `rolling`, the outcome of the roll is stored in `roll`. Assume an equally likely occurrence of each possible outcome of a roll.

(e) **[3 marks]** Write a function `printRoll` that takes in the `roll` array and displays the roll with each name separated by a single space.

(f) **[3 marks]** Write a function `getPayout` that takes in the number of coins played (`numCoins`), the arrays `roll` and `reel`, and returns the payout.

(g) **[9 marks]** Complete the `main` function to allow a gambler to play jackpot. Each time the gambler is required to place a bet of between 1 and 5 coins inclusive. The program repeatedly asks the gambler if he/she wishes to continue after each roll and payout. The program terminates when an `N` is entered. The sample run of the program is shown on the next page.

```
Insert coins: 0
You have inserted an invalid number of coins. Please re-enter.

Insert coins: 3
You have inserted 3 coins.
Rolling... Lion Flower 3
Your payout is 0 coins.
Do you want to continue playing? (Y/N) Y

Insert coins: 4
You have inserted 4 coins.
Rolling... 3 3 3
Your payout is 16 coins.
Do you want to continue playing? (Y/N) N

Thank you for playing. Goodbye.
```

ANSWER:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <time.h>
```

```
/* Part (a) */
```

```
/* Function Prototypes */
void rolling(struct Object *roll[],
             struct Object reel[]);

int getPayout(int numCoins,
              struct Object *roll[],
              struct Object reel[]);

void printRoll(struct Object *roll[]);
```

```
int main(void)
{
    /* Part (b) */

    /* Part (c) */

   int numCoins, payout;  // number of coins played and payout
   char ans[2];           // assume only Y or N response

   srand((unsigned int)time(NULL));

    /* Part (g) */

   return 0;
}
```

```
/* Part (d) */
```

```
/* Part (e) */
```

```
/* Part (f) */
```

— END OF PAPER —