NANYANG TECHNOLOGICAL UNIVERSITY

SEMESTER 2 EXAMINATION 2011-2012

CE1007/CZ1007 – DATA STRUCTURES

April/May 2012 Time Allowed: 2 hours

INSTRUCTIONS

1. This paper contains 4 questions and comprises 5 pages.

2. Answer **ALL** questions.

3. This is a closed-book examination.

4. All questions carry equal marks.

---

1. (a) State the output produced by the `printf()` statements of the program in Figure Q1.

```
#include <stdio.h>
void f(int x, int *y);
int main( )
{
   int h=8, k=0;
   f(h, &k);
   printf("h = %d, k = %d\n", h, k);
   return 0;
}
void f(int x, int *y)
{
   printf("x = %d, *y = %d\n", x, *y);
   if (x > 0)  {
      *y += x;
      f(x-2, y);
   }
   printf("x = %d, *y = %d\n", x, *y);
}
```

Figure Q1

(11 marks)

Note: Question No. 1 continues on Page 2

(b) Give one advantage of using iteration instead of recursion and one advantage of recursion instead of iteration. Explain your answers.

(4 marks)

(c) Write a recursive C function *digitPos* that takes a positive integer argument *n* and returns the position of the first appearance of a specified *digit* in the integer *n*. The position of the digit is counted from the right and starts from 1. If the required digit is not in the number, the function should return 0. For example, *digitPos(42345, 4)* returns 2 and *digitPos(123, 6)* returns 0. No error checking on the parameters is required in the function. In addition, you must not use an **array** or a **string**. The prototype of the function is given below:

```
int digitPos(int n, int digit);
```

(10 marks)

2. (a) What are the differences and similarities between an array and a string in C programming?

(3 marks)

(b) Write a C function *findString* that takes two character string arguments, *s* and *t* as input and returns 1 if *s* is a substring of *t* (i.e. if *s* is contained in *t*) and -1 if not. For example, the function will return 1 if *s* is "123" and *t* is "abc123xyz", but it will return -1 if *s* is "123" and *t* is "abc12xyz". You must not use any of the string functions from the standard C libraries. The prototype of the function is given below:

```
void findString(char *s, char *t);
```

(10 marks)

(c) Determine the output of the program given in Figure Q2, where the addresses of `ch`, `ptr1`, `ptr2` and `ptr3` are 0022FF0B, 0022FEE4, 0022FEE0 and 0022FEDC respectively.

Note: Figure Q2 is on Page 3

```
#include <stdio.h>
int main( )
{
   char ch;
   char ar1[4][6] = {
                   {'P', 'e', 't', 'e', 'r'},
                   {'J', 'o', 'h', 'n', 'n', 'y'},
                   {'M', 'a', 'r', 'y'},
                   {'K', 'e', 'n', 'n', 'y'}
                    };
   char *ar2;
   char *ar3[2];
   char *ptr1, **ptr2, ***ptr3;

   ch = 'a';
   ptr1 = &ch; ptr2 = &ptr1; ptr3 = &ptr2;
   printf("Output1 = %p\n", &ptr2);
   printf("Output2 = %p\n", **ptr3);
   printf("Output3 = %p\n", *ptr3);
   printf("Output4 = %p\n", *ptr2);

   ar2 = &ar1[1][1];
   ar3[0] = &ar1[0][0];
   ar3[1] = &ar1[1][0];
   printf("Output5 = %s\n", &ar1[1][4]);
   printf("Output6 = %s\n", ar2-3);
   printf("Output7 = %s\n", ar3[0]+1);
   printf("Output8 = %s\n", ar3[1]+2);
   return 0;
}
```

Figure Q2

(12 marks)

3. (a) Explain the importance of `malloc()` and `free()` in the implementation of dynamic data structures in C programming.

(3 marks)

(b) Consider the code given in Figure Q3 for the C structure defining a node in a singly-linked list of integers.

Note: Question No. 3 continues on Page 4

```
struct ListNode{
  int item;
  struct ListNode *next;
};
```

Figure Q3

Modify the `ListNode` structure to support the following requirements:

I. The linked list should be able to store any form of data in its nodes instead of just integer values.

II. Given a data item "*X*" and an integer *P*, the linked list should support efficient insertion of a copy of the node *N* containing item "*X*", in the position *P* nodes before *X*. You may assume that the new position exists and "*X*" can be found.

For example, if the string "*Test Item*" is the data item in the seventh node, and *P = 3*, a copy of the seventh node *N* should be inserted in the fourth position of the linked list, and *N* will now become the eighth node.

(7 marks)

(c) A bank has asked you to implement an application for keeping track of customer records and banking activities. While the application is running, customer records should be maintained in ascending alphabetically sorted order (A-Z). Each customer may have any number of accounts with the bank; the bank provides both checking and savings accounts. Each account should maintain a record of all transactions carried out since the opening of that account.

When designing your application, you should consider that this bank is the only financial institution in its area. This means that its customer list rarely changes and customers infrequently sign up for new accounts, although many transactions are made every day.

Using appropriate diagrams, state and justify the data structure (or combinations of data structures) used in your application.

(15 marks)

4. (a) Draw the Binary Search Tree generated after inserting the following numbers in the order that they appear:

    99, 4, 19, 36, 64, 30, 3, 1

(6 marks)

(b) Does the tree you generated in Q4(a) allow for efficient search? Explain your answer.

(7 marks)

(c) Write the pseudocode for a function *IsBST* which checks whether a given Binary Tree B is also a Binary Search Tree.

(12 marks)

END OF PAPER

CE1007 DATA STRUCTURES
CZ1007 DATA STRUCTURES

Please read the following instructions carefully:

1. **Please do not turn over the question paper until you are told to do so. Disciplinary action may be taken against you if you do so.**

2. You are not allowed to leave the examination hall unless accompanied by an invigilator. You may raise your hand if you need to communicate with the invigilator.

3. Please write your Matriculation Number on the front of the answer book.

4. Please indicate clearly in the answer book (at the appropriate place) if you are continuing the answer to a question elsewhere in the book.