**NANYANG TECHNOLOGICAL UNIVERSITY**

**SEMESTER 2 EXAMINATION 2012-2013**

**CE1007/CZ1007 – DATA STRUCTURES**

Apr/May 2013 Time Allowed: 2 hours

**INSTRUCTIONS**

1. This paper contains 4 questions and comprises 4 pages.
2. Answer **ALL** questions.
3. This is a closed-book examination.
4. All questions carry equal marks.

---

1. (a) What is the output of the following code?

```
#include <stdio.h>
   int f(int n);
   int main()
   {
      printf("Result = %d\n", f(6));
      return 0;
   }
   int f(int n)
   {
      int a, b;
      if (n>2){
         a = f(n-1);
         b = f(n-2);
         printf("a = %d, b = %d\n", a, b);
         return a+b;
      }
      else
         return 2;
   }
```

(8 marks)

Note: Question No. 1 continues on Page 2

(b) Given the following C declarations:

```
struct student {
    int age;
    char *name;
};
struct student b[3] = {
    18, "Peter", 19, "Mary", 20, "John"
};
struct student *p = b;
```

Assume that the address of the first element of the array `b[3]` is 1010 and the following operations are executed sequentially, in order, from (i) to (vi). What are the values of the following expressions?

(i) `++p->age`
(ii) `(*p).name`
(iii) `*p->name-1`
(iv) `*++p->name`
(v) `*p++->name`
(vi) `*(++p)->name`

(7 marks)

(c) Write a **recursive** C function `reverseAr()` that reverses the contents of an array. The function takes in two arguments `ar[]` and `n`, which are an array of characters and an integer specifying the size of the array respectively. For example, if the array contains, in order, 1, 2, 3, 4, 5, then after reversal, its content should be, in order, 5, 4, 3, 2, 1. The code should not use any other arrays. The function prototype is given as follows:

```
void reverseAr(char ar[], int n);
```

(10 marks)

2. (a) Write a C function `squeeze()` that removes any occurrences of a given character `c` from the string `str`. For example, if the string content is `"abcdabcd"` and the character is `'b'`, then after removal, its content should be `"acdacd"`. The code should not use any of the standard string library functions. The function prototype is given as follows:

```
void squeeze(char str[], char c);
```

(7 marks)

Note: Question No. 2 continues on Page 3

(b) The C function `strcmp()` compares the string pointed to by `s1` to the string pointed to by `s2`. If the string pointed to by `s1` is greater than, equal to, or less than the string pointed to by `s2`, then it returns 1, 0 or –1 respectively. Write the code for the function without using any of the standard string library functions. The function prototype is given as follows:

```
int strcmp(char *s1, char *s2);
```

(9 marks)

(c) Given the following C declaration:

```
char *a[2][3] = {"abc", "defghi", "ijkl",
                 "mnopqr", "stuvm", "xyz"};
```

Assume that the address of the first element of the array is 1010 and the following operations are executed sequentially, in order, from (i) to (v). What are the values of the following expressions?

(i) `***(a+1)`
(ii) `**a[0]`
(iii) `(*(*(a+1)+1))[4]`
(iv) `*(a[1][2]+2)`
(v) `**(a+1)`

(9 marks)

3. (a) Explain how a memory leak occurs in C programs and why it is undesirable.

(6 marks)

(b) Write a C function `exchange()` that traverses a linked list of integers at most once, finds the nodes storing the maximum and minimum values in the list, and swaps those nodes. You may assume that all values stored in the list are unique. The function prototype is given as follows:

```
void exchange(linkedlist *ll);
```

For example, calling `exchange()` on the list [28 15 8 6 2 14] would result in the modified list [2 15 8 6 28 14].

(12 marks)

Note: Question No. 3 continues on Page 4

(c) A queue can be simulated using two stacks. Using appropriate diagrams and/or pseudocode, describe how the `enqueue()` and `dequeue()` functions would be implemented for this simulated queue.

(7 marks)

4. (a) State two differences between the linked list and binary tree data structures.

(6 marks)

(b) Draw all possible minimal-height Binary Search Trees that store the list *16, 44, 99, 1*.

(10 marks)

(c) Write a C function `countonechild()` that accepts a pointer to the root node of a binary tree and returns the number of nodes with exactly one child node. The function prototype is given as follows:

```
int countonechild(btnode *node);
```

(9 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.