NATIONAL UNIVERSITY OF SINGAPORE

UIT2201 : COMPUTER SCIENCE AND THE IT REVOLUTION

(Semester II : AY 2011-12)

April 2012 – Time Allowed: 2 Hours

INSTRUCTIONS TO CANDIDATES

1. This examination paper consists of **FIVE** questions and comprises **TEN** printed pages including this page.
2. Answer **ALL** questions.
3. **Write ALL your answers in this examination book.**
4. This is an **OPEN BOOK** examination.

Matric. Number: _____

QUESTION	POSSIBLE	SCORE
Q1	20	
Q2	15	
Q3	15	
Q4	15	
Q5	15	
TOTAL	**80**	

Question 1: (20 marks)

True-False questions. (2 marks each)

(a) Algorithms *were first developed* even before the first general purpose electronic computers were built in the 1950s. _____

(b) A person who carries out the steps contained in an algorithm *does not need to understand* the concepts or ideas underlying the algorithm. _____

(c) The binary search algorithm can still be used when the array is *unsorted*, but it will be slightly slower. _____

(d) Suppose $\boldsymbol{D} = (D_1, D_2, \ldots, D_n)$ is a list of number, that is *nearly sorted* (meaning that only a few numbers in $\boldsymbol{D}$ are *out of place*, but we *do not know* which ones are out of place). Sorting the list $\boldsymbol{D}$ using the *selection sort algorithm* takes *linear time,* $\Theta(n)$. _____

(e) The database query processing language, SQL, is an example of a *procedural language* in which the user specifies the *procedure used to compute* the answer of the user queries. _____

(f) Any real number that can be represented in base 10 can also be *represented exactly* in base 16 (hexadecimal). _____

(g) We can build a logic circuit for any 4-variable logic function (for example a logic function with input variables A, B, C and D) using *only* XOR gates and NOT gates. _____

(h) The idea of *abstraction* can be applied to *both* software (algorithm design and development) and hardware (design of digital circuits). _____

(i) A design feature of the ALU (Arithmetic/Logic Unit) is that *all of the circuits will compute their different operations* on the inputs, and then the desired output will be chosen from among them. _____

(j) Given the rule "if X is smiling, then X is friendly", and the assertion "Tom is friendly", one can conclude that "Tom is smiling". _____

Fun Question: (1 bonus mark)

In the joke (told in class)

"There are **10** types of mathematicians in this world:
those who can think *binarily*, (is this a word?) and those who can't.",

explain why **10**.

Your Answer: _____

Question 2: (15 marks)

Consider a database with the following 3 tables: **{SI, CI, EN}**. We assume that |**SI**|=30,000, |**CI**|=1000, |**EN**|=100,000. (To save space and writing, you should use the short table names.)

SI (STUDENT-INFO)						
Student-ID	**Name**	**NRIC-No**	**Address**	**Tel-No**	**Faculty**	**Major**
---	---	---	---	---	---	---

CI (COURSE-INFO)					
Course-ID	**Name**	**Day**	**Hour**	**Venue**	**Instructor**
---	---	---	---	---	---

EN (ENROLMENT)	
Student-ID	**Course-ID**
---	---

We want to obtain the list of students who major in **"Origami"** (a *newly introduced major*) and are enrolled in the course with **Course-ID "UIT2201"**. For these students, we want a list that contains their **Student-ID, Name, Tel-No**.

(a) (3 marks) Give an appropriate **SQL query** to accomplish this task.

(b) (3 marks) One simple method for the DBMS to satisfy the query is to first perform **e-join** operations to join the three tables **{SI, CI, EN}**, and then selecting (using **e-select**) and projecting (using **e-project**) the answer.
Explain why this method is ***not the preferred method*** to accomplish this task.

Question 2: (continued…)

(c) (3 marks) Give an appropriate sequence of ***basic database primitives*** operations (using **`e-project, e-select, e-join`**) to accomplish this task.

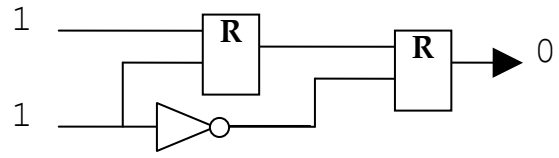
Suppose that the three tables **`{SI, CI, EN}`** were suitably combined into one big table called **`ALL`** (by using the appropriate **`e-join`** operations). Then it will be easy to answer *any query* using this giant table **`ALL`**.

(d) (2 marks) One problem with this database scheme with a giant table **`ALL`** is ***the duplication of data***. Explain clearly what *duplication of data* means in this context.

(e) (4 marks) There is ***a more serious problem*** with this database scheme with a giant table. What is this more serious problem and explain how/why it happens.

Question 3: (15 marks)

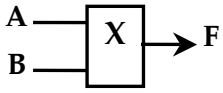
(a) (3 marks) In the circuit below, the rectangles **R** represent the same type of gate. Based on the input and output information given, identify whether **R** can be any one of the following: AND, OR, or XOR gate.



Circle the correct answer below:

AND-gate: Yes No **OR-gate:** Yes No **XOR-gate:** Yes No

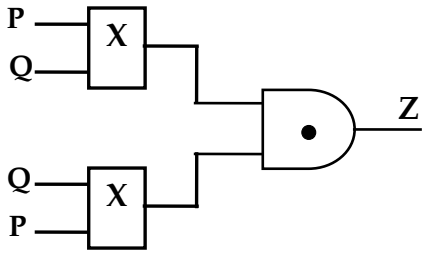
(b) (4 marks) In your logic design laboratory, you are given a "*mystery*" *gate* **X** (shown below) with the truth table defined on the right.



A	B	F
0	0	***1***
0	1	***1***
1	0	***0***
1	1	***1***

Using a truth table (or otherwise), give the logical formula for the output **Z** of the circuit shown below.

P	Q	Z		
0	0			
0	1			
1	0			
1	1			



Answer: **Z** = _____

Question 3: (continued…)

You are given a random access memory (RAM) unit that is square in shape and where the MAR is 32 bits long and the MDR is 16 bits long.

(c) (2 marks) What is the size of the memory unit? (You can leave your answer as a power of 2, if you wish.)

Answer: _____

(d) (2 points) How many bits are there in the *row selector* and how many in the *column selector*?

Row Selector: _____ **Column Selector:** _____

(e) (4 points) The phenomena of "*repeated doubling / halving*" occurs in many places in computing. Give an example of this in **(i)** ***algorithm design***, and an example in ***(ii) binary digital circuit and computer design***, with a short explanation.

Answer: (i) … in algorithm design

Answer: (i) … in binary digital circuit and computer design

Question 4: (15 marks)

Two startup companies, **MeSM** and **MeBG**, recently merged and form **MeCool**. **MeSM** has a small customer list **SM** with 1,000 customers and uses *sequential search* algorithm to lookup its customers. On the other hand, **MeBG** has a larger customer list **BG** with 1,000,000 customers and uses *binary search* algorithm to lookup its customers. The new boss at **MeCool** wants the ICT department to quickly implement a new lookup function for the combined customer list.

As a quick-and-dirty solution for the lookup function, the software developer at **MeCool** ICT department proposed the following algorithm (in pseudo-code):

> **Algorithm Hybrid-SM-then-BG (q, SM, BG)** **(* q is a given name *)**
> 1. Use *sequential search* to search for `q` in the list `SM`;
> 2. If unsuccessful, use *binary search* to search for `q` in the list `BG`;

Before implementing the proposed solution, **MeCool** quickly hired you as an ICT consultant to analyze and give your advice on the proposed algorithm.

(a) (2 marks) Give the **main advantage** of implementing the proposed solution.

(b) (4 marks) When searching (*successfully*) for *a customer from the list* **SM**, what is the ***number of comparisons*** in the **worst-case** and **average-case**?
(For average case, assume that all names are *equally likely* to be searched.)

Worst Case: _____ **Average Case:** _____

(c) (3 marks) When searching (*successfully*) for *a customer from the combined list, namely,* (**SM** + **BG**), what is the ***number of comparisons*** in the **worst-case**?

Worst Case: _____

Question 4: (continued…)

(d) (4 marks) As the ICT *consultant*, you want to propose an algorithm for **MeCool** that is *much better* than the quick-and-dirty proposed solution. Propose *your best* algorithm for searching the combined list. Informally, describe your algorithm using high-level pseudo-code similar to that given above.
[Do NOT give detailed code for this problem.]

(e) (2 marks) When searching (*successfully*) for *a customer from the combined list* (**SM** + **BG**), what is the ***number of comparisons*** needed in the *worst-case* ***using your algorithm*** proposed in **(d)** above?

Worst Case: _____

Question 5: (15 marks)

You are given a knowledge based system with the following knowledge base on a road network with road junctions **A, B, C, D, E**.

Knowledge/Fact Base:

```
ROAD(A,B),    ROAD(A,D),
ROAD(B,C),    ROAD(B,D),
ROAD(C,B),
ROAD(D,E),
ROAD(E,C),    ROAD(E,D)
```

(Note: Here, **Road(X,Y)** represents a Road from **X** to **Y**.)

Inference Rules:

```
R1.   PATH(X,Y) if ROAD(X,Y)                 // base case
R2.   PATH(X,Y) if ROAD(X,Z) and PATH(Z,Y)   // recursive case
```

(a) (3 marks) Draw the "road network" based on the fact base given.

(b) (5 marks) Answer the following queries: (no need to show the steps)

?ROAD(B,C) **Answer:** _____

?ROAD(**X**,D) **X =** _____

?PATH(A,B) **Answer:** _____

?PATH(A,E) **Answer:** _____

_____

_____

Question 5: (continued…)

You are given a list $A[1..n]$ of n positive integers. (For simplicity, you can assume that the integers are all distinct.) We want to choose two numbers that has the ***maximum sum***. (For example, if $A = [9, 1, 3, 4, 2, 7]$, the two numbers are 7 and 9 and the maximum sum is 16.)

(c) (2 points) Design an algorithm for computing this maximum sum. Give your algorithm in pseudo-code. (You are free to quote any algorithm covered in the course. Quote them as *high-level primitives* and clearly state what they do.)

Now, suppose that you want to choose two numbers from $A[1..n]$ that sum up to a given target number X. (In the above example, if the target number X=11, then we choose numbers 4 and 7 to give a sum of 11.)

(d) (5 points) ***Design an algorithm*** for choosing two numbers from the list $A[1..n]$ that sum up to a given target number X. Give your algorithm in pseudo-code. (You are free to quote any algorithm covered in the course. Quote them as *high-level primitives* and clearly state what they do.)

~~~ END OF QUESTIONS ~~~