109cc01

國立臺北科技大學 109 學年度碩士班招生考試

系所組別：2210 電子工程系碩士班甲組

第一節　計算機概論　試題

注意事項：

1. 本試題共 4 題，每題及小題配分列於題目後，共 100 分。
2. 不必抄題，作答時請將試題題號及答案依照順序寫在答案卷上。
3. 全部答案均須在答案卷之答案欄內作答，否則不予計分。
4. 所有題目均請詳述解答過程。

一、簡答題：(100%)

(一) Computer Programming Techniques. (25%)

1. Implement a function `unsigned int myAvg(unsigned int x, unsigned int y)` to return the *truncated average* of two given unsigned integers `x` and `y`. Note that the *truncated average* discards all values less than 1, so `myAvg(3, 8)` should return `5`. Your code must consider the potential *overflows* and should be as efficient as possible. (10%)
2. Implement a function `bool isPowerOf2(unsigned int x)` to determine whether a given unsigned integer `x` is a power of 2. You should implement the function from scratch without using STL or other libraries. Your code should be as efficient as possible, in terms of both the running time and memory space usage. (15%)

(二) Data Structures: Stack. (15%)

1. Please use C or C++ programming language to implement an *array-based stack*, which accepts up to N elements. Each element is an unsigned integer. You should implement the stack from scratch, without using STL or other libraries. (5%)
2. Based on the answer of the previous problem, extend your code so that your stack can support an extra operation `int getMax()`, which returns the largest element in the stack *without removing it*. (10%)

(三) Data Structures: Search Tree. (25%)

1. As compared to *binary search tree (BST)* and *red–black tree*, explain the major design ideas of *B-tree*. (5%)
2. Define and explain the five *red-black properties*. (10%)
3. Prove that each 2–3–4 tree can be converted to an equivalent red–black tree, and vice versa. (10%)

(四) Computer Organization and Operating Systems. (35%)

1. Define *Amdahl's law* and its implications on the performance optimization of computer systems. (5%)
2. Define and differentiate the two terms: *parallelism* and *concurrency*. Why are they important for the performance of modern computers? (5%)
3. Explain the *pipelining* technique in CPU design. (5%)
4. Explain the *non-local jumps* in C language. Why are they needed? (5%)
5. Explain how *signals* occur and how they are handled. (10%)
6. What is the *signal mask*, and why is it needed? (5%)