一. 选择题 (30%)

1. ( ) The markup language used in WAP is called (a) HTML (b) GML (c) SGML (d) WML (e) none of the above.

2. ( ) When you _____, you send the address of a variable to the called function rather than sending its contents. (a) write a function declaration (b) pass by value (c) use a return statement (d) pass by reference (e) none of the above.

3. ( ) The idea of hiding the data and some operations that can be performed on the data inside the object is called _____. (a) containment (b) translating (c) encapsulation (d) encryption (e) none of the above.

4. ( ) If you were looking up a name in a phone book, you would use a _____. (a) selection sort (b) insertion sort (c) binary search (d) sequential search (e) none of the above.

5. ( ) Algorithm complexity _____. (a) helps to explain how an algorithm is coded (b) helps to decide using a searching algorithm or a sorting algorithm (c) is a measurement that is used to decide which algorithm to use in a given situation (d) is to be avoided (e) none of the above.

6. ( ) A recursive solution often replaces a(n) _____ with a(n) ____. (a) loop, decision (b) iteration, repetition (c) sequence, decision (d) loop, sequence (e) none of the above.

7. ( ) In a(n) _____ sort, each number is inserted into the proper place in the list to maintain a particular sequence. (a) insertion (b) bubble (c) sequential (d) selection (e) none of the above.

8. ( ) Starvation can happen when the OS _____. (a) puts a process into a hold state (b) puts too many resource restrictions on a process (c) puts too few restrictions on processes and resources (d) puts a process into a terminated state. It is not a form of starvation. (e) none of the above.

9. ( ) The relationship between a program, a job, and a process can be illustrated with a(n) _____ that shows the different states of each. (a) process scheduler (b) block diagram (c) state diagram (d) Job scheduler (e) none of the above.

10. ( ) In _____, the program is divided into pages, but the pages can be loaded into memory one by one, executed, and replaced by another page. (a) segmentation (b) demand segmentation (c) virtual memory (d) demand paging (e) none of the above.

11. ( ) The presentation layer is concerned with the ____ and semantics of the information exchanged between two systems. (a) punctuation (b) syntax (c) appearance (d) route (e) none of the above.

12. ( ) A(n) _____ is a traffic controller that can divide a long bus into smaller segments. (a) hub (b) bridge (c) gateway (d) server (e) none of the above.

13. ( ) Data retrieval speed can be increased dramatically by using _____. (a) indexes (b) pointers (c) flags (d) SQL (e) none of the above.

14. ( ) The two properties of time stamping method is uniqueness and _____. (a) atomicity (b) atomic (c) isolation (d) monotonicity (e) none of the above.

15. ( ) The mobile device having the capability of cell phone’s voice communication and PDA’s PIM is called (a) Smartphone (b) Wonderphone (c) E-phone (d) M-phone (e) none of the above.
二．問答題和程式題

1. Two of the main techniques for re-using code are “class inheritance” and “object composition”. Please describe the advantages and disadvantages of each technique. (10%)

2. T0 and T1 are two concurrent executing transactions. Time is running down from top to bottom.
   (a) What is serializability? (6%)
   (b) Is the schedule at right a serializable schedule?
       Justify your answer. (6%)

<table>
<thead>
<tr>
<th>T0</th>
<th>T1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read (A)</td>
<td>Read (A)</td>
</tr>
<tr>
<td>Write (A)</td>
<td>Read (B)</td>
</tr>
<tr>
<td>Read (B)</td>
<td>Write (A)</td>
</tr>
<tr>
<td>Write (B)</td>
<td>Write (B)</td>
</tr>
</tbody>
</table>

3. The running time of quicksort depends on whether the partitioning is balanced or unbalanced, and this in turn depends on which elements are used for partitioning. Answer the following questions in terms of Big-O notation.
   (a) Prove that if the partitioning is balanced, the quicksort algorithm runs asymptotically as fast as merge sort. (6%)
   (b) Prove that if the partitioning is unbalanced, however, quicksort can run asymptotically as slow as insertion sort. (6%)

4. Maze is widely used in computer games. A maze is typically represented by a two dimensional array of integers. For example, the following maze uses 1 to indicate a clear path and 0 a blocked path. Use your favorite language or pseudo code to implement the following traverse function. (12%)

   // Return
   // true if there is a path from “Start” to “Goal”
   // false, otherwise
   boolean traverse (int row, int column) {
       // you can invent any functions you need as long as
       // they are clearly explained.
       // For example, valid(row, column) returns true if the cell at row, column is movable.
       return
   }
5. Regarding Mobile IP,
   (a) What is Mobile IP? (6%)
   (b) How does its roaming work? (6%)

6. Regarding the Java programs shown below,
   (a) What is the output of the main program TestCircularList.java? (6%)
   (b) Implement the class “Queue” using class “CircularList”,

   ```java
   class Queue extends CircularList {
     public Queue () { // needs to be implemented }
     public void enterq (int x) { // needs to be implemented }
     public int leaveq () { // needs to be implemented }
     public boolean empty() { // needs to be implemented }
   }
   ```

   // Main program: TestCircularList.java
   ```java
   import java.io.*;
   import java.lang.*;
   public class TestCircularList {
     public static void main(String args[]) {
       CircularList cl = new CircularList();
       cl.push(10);
       cl.push(12);
       cl.enter(14);
       cl.pop();
       cl.enter(14);
       cl.push(16);
       cl.push(19);
       int num = cl.size();
       for (int i=0; i<num; i++)
         System.out.println(cl.pop());
     }
   }
   ```

   // Supporting class: CircularList.java
   ```java
   class CircularList {
     Cell rear;
     int size;
     public CircularList() {
       rear = new Cell(0);
       size = 0;
     }
     public boolean empty() {
       return rear == rear.next();
     }
     public void push(int x) {
       rear.next = new Cell(x, rear.next);
       ++size;
     }
     public int pop() {
       if (empty())
         return 0;
       size--;
       Cell front = rear.next;
       rear.next = front.next;
       int x = front.info;
       front = null;
       return x;
     }
     public void enter(int x) {
       ++size;
       rear.info = x;
       rear = rear.next = new Cell(0, rear.next);
     }
     public int size() {
       return size;
     }
   }
   ```

   // Supporting class: Cell.java
   ```java
   class Cell {
     int info;
     Cell next;
     public Cell(int i) {
       info = i;
       next = this;
     }
     public Cell(int i, Cell n) {
       info = i;
       next = n;
     }
     public Cell next() {
       return next;
     }
   }
   ```