※ Please detail your answer for each question in the answer sheet.

1. **Multiple-choice questions (total 20%; each subquestion deserves 4%):**

   There may be **ZERO or MORE correct answers** in each subquestion. If there is no correct answer, please give the answer "None".

   (1) Build a binary search tree for the input sequence 12, 5, 7, 18, 19, 15, 14, 6, 13. It is assumed that the tree root is on level 1. (A) There are five levels in the tree. (B) 14 is on level 3. (C) If we want to search 16, the number of required node comparisons is 3. (D) The subtree rooted at 15 has 3 nodes (including 15).

   (2) Which statement(s) is correct for an AVL tree? (A) The absolute value of the level difference of any two leaves is at most one. (B) The absolute value of the height difference of any two subtrees on the same level is at most one. (C) A deletion needs at most two rotation operations to preserve an AVL tree to be a height-balanced tree. (D) After a new node is inserted, the tree height will not increase if rotation operations are performed.

   (3) In a stack structure, let u denote a push operation and o denote a pop operation. After a sequence of stack operations, consisting of 3 pushes and 3 pops, an input sequence 123 may change its order. For example, after uouuoo is performed, 123 will become 132. Which sequence(s) cannot be obtained from input 123 by stack operations? (A) 213 (B) 231 (C) 312 (D) 321.

   (4) Which statement(s) is correct for binary search on a sequence? (A) The sequence must be sorted. (B) Binary search can be implemented by an array. (C) Binary search can be implemented by a linked list. (D) The hashing technique is an implementation of binary search.

   (5) Which statement(s) is correct for arithmetic expressions? (A) There is no parenthesis in a postfix expression. (B) The first symbol in a prefix expression must be an operator. (C) The first symbol in a postfix expression must be an operator. (D) The amount of operands in a prefix is more than that of operators.

2. **Printed results in C codes (total 10%; each subquestion deserves 5%):**

   What are printed in each of the following program segments?

   (1)
   ```
   int a[ ]={5,7,9,11,13};
   int *p;
   int i=2;
   p=a;
   *(p++) = ++i;
   printf("%d %d %d %d %d",a[0],a[1],a[i++],*p,*(p+1));
   ```

   (2)
   ```
   int b=26;
   printf("%d", ((b >> 3) << 2) & ((b <<2) >> 3));
   ```

3. **Two-way insertion sort (total 10%):**

   Suppose the output sorted sequence is increasing. The two-way insertion sort is a modification of the simple insertion sort (straightforward insertion sort), described as follows. After read the input elements, a separate output array a[0], a[1], a[2], ... a[n-1] is used to store the sorted sequence. This output array acts as a circular structure, that is, the right position of a[i] is a[i+1] if 0 ≤ i ≤ n-2, and the right position of a[n-1] is a[0]. The first input element is put into a[0] initially. Once a contiguous group of elements are in the array, room for a new input element is made by shifting all smaller

elements one step to the left or all larger element one step to the right. The choice of left-shift or right-shift to perform depends on which would cause the smallest amount of shifts. Use the following 6 input elements to illustrate how this algorithm works. You have to show the content of the output array after each input element is inserted into the array.

27, 35, 43, 31, 29, 33.

4. Huffman tree (total 10%):

Huffman algorithm is a variable-length encoding method, in which a Huffman tree is constructed. Suppose seven symbols A, B, C, D, E, F, G with frequencies 8, 15, 1, 6, 15, 3, 7, respectively, are given. You have to obey the following rules for constructing the Huffman tree:

(1) The label of each leaf node is a single symbol. The label of each internal node is the concatenation of the labels of its left child and right child.

(2) When two nodes are merged, always set the node of label with less lexical order as the left child, and the other as the right child.

(3) If more than two nodes have the same least frequency, then you have to merge the two that have the least and the second least label in lexical order.

Please draw the full Huffman tree.

5. True-false questions (total 14%; each subquestion deserves 2%):

Please mark either 'O' or 'X' to respectively answer "yes" or "no" for each subquestion. Notice that you will get **-2 point** for each wrong answer until no point can be deducted from this question.

(1) The need-to-know principle dictates that programs, users, and even systems should be given just sufficient privileges to perform their tasks. This principle is very critical for system protection, which can prevent the mischievous, intentional violation of an access restriction by a user.

(2) A deadlock situation must arise if any of the following four conditions holds in a computer system: mutual exclusion, hold and wait, no preemption, and circular wait. Therefore, to vanquish the deadlock, we have to see what conditions cause that deadlock and then try to conquer each of them.

(3) We can realize a dynamic relocation by using a relocation register in memory management. For example, suppose that the CPU asks a logical address of 123 and the relocation register has a value of 1877. Then, the physical address will be 2000.

(4) A trap door may be included in a compiler. Therefore, the compiler would generate the standard object code as well as a trap door. However, this activity can be still found by conducting a very careful search of the source code of the compiled program.

(5) An attack that one participant in a communication pretends to be someone else is called the man-in-the-middle attack. Such an attack breaches the correctness of identification. A cracker can use this attack to gain access that he/she would not normally be allowed or escalate his/her privileges.

(6) RAID level 0+1 refers to a combination of RAID levels 0 and 1, where RAID 0 provides the performance, while RAID 1 supports the reliability. Generally speaking, this level provides better performance than RAID 5. It is common in environments where both performance and reliability are important.

(7) Location independence and location transparency are two related notions regarding name mappings in a distributed file system. A location-transparency naming scheme is a dynamic mapping, since it can map the same file name to different locations at two different times.

Therefore, location transparency is a stronger property than location independence.

6. **Choice questions (total 10%; each subquestion deserves 2%):**

Please select the **BEST ONE ANSWER** for each subquestion from the following box:

A.	page fault	B.	process	C.	page replacement
D.	worm	E.	virus	F.	authentication
G.	Trojan horse	H.	encryption	I.	class
J.	bnode	K.	demand paging	L.	inode
M.	thread	N.	cnode	O.	decryption

(1) The smallest unit of CPU utilization, which can be executed independently by OS

(2) The technique that loads pages only as they are needed

(3) A code fragment embedded in a legitimate program, which is designed to self-replicate & infect other programs

(4) The security technique that supports nonrepudiation

(5) A file-control block in most UNIX file systems

7. **Disk scheduling (total 6%):**

Suppose that you are given a disk queue with requests for I/O to blocks on cylinders 98, 183, 37, 122, 14, 124, 65, and 67. Let the disk head start at cylinder 53. Please show the disk scheduling result by

(1) The shortest-seek-time-first algorithm (3%)

(2) The SCAN algorithm (3%)

8. **Distributed synchronization (total 10%):**

(1) In a distributed system, deciding the event ordering is usually difficult. Why? (2%)

(2) Let A, B, C be events and '→' denote the happen-before relation between two events. Please explain the three principles of the happen-before relation in a distributed system using the above notations. (6%)

(3) What is an atomic transaction? (2%)

9. **Memory management (total 10%):**

(1) What is a reentrant code? (2%)

(2) How does the CPU know whether a page is available or not from the page table? (2%)

(3) Suppose that it takes 20 nanoseconds to search the TLB (translation look-aside buffer) and 100 nanoseconds to access memory. If we want the effective memory-access time no longer than 122 nanoseconds, what is the hit ratio for the TLB? (6%)