1. [15] Suppose we have a byte-addressable machine, i.e., each byte is accessed via an address. Let the locations of an array be allocated in a row-major manner. In a program segment, we declare an array *PP* by

   $$int\ PP[20,30];$$

   Let each integer take 4 bytes and the location of this array start with address 1000. Please answer the following questions:
   1.1 [3] How many integers can be stored in the array *PP*?
   1.2 [3] How many bytes does the array *PP* itself take?
   1.3 [3] What is the starting address of the element *PP[10,5]*?
   1.4 [3] What is the starting address of the element *PP[5,10]*?
   1.5 [3] What is the starting address of the last element in *PP*?

2. [15] Suppose we use a singly linked list to store integers in its nodes. Assume that, initially, 100 is stored in the first node starting at address 2000, 200 is stored in the second node starting at address 500, 300 is stored in the third node starting at address 1000, and 400 is stored in the fourth node starting at address 100. Let the starting address of the first node of the list is stored in the variable named head. Let an integer take 4 bytes and a pointer (or address) take 8 bytes. Please answer the following questions:
   2.1 [3] What is the size, in the number of bytes, of each node of the list?
   2.2 [3] What is stored in the variable head?
   2.3 [3] What is the content of the first node of the list?
   2.4 [3] What is the content of the last node of the list?
   2.5 [3] Suppose we delete the second node from the list. After deletion, what is the content of the first node of the resulting list?

3. [15] Usually, there are three ways to display the nodes of a binary tree: preorder, inorder, and postorder. Consider Figure 1 and answer the following questions:
   3.1 [5] What is the fifth node of the preorder sequence of this tree?
   3.2 [5] What is the fifth node of the inorder sequence of this tree?
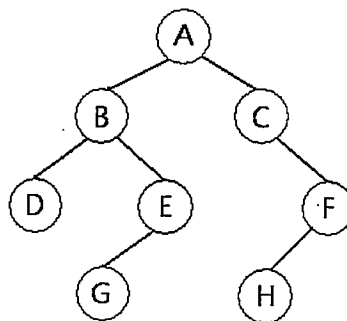   3.3 [5] What is the fifth node of the postorder sequence of this tree?

Figure 1

背面有題

4. [15] Initially, the array *AA* shown in Figure 2 stores a max heap with 12 integers. Note that A[0] is not used. Suppose we Insert 105 into the heap and then make it into a max heap.

	120	90	100	65	20	80	60	55	25	10	5	40			
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Figure 2

Please answer the following questions:
4.1 [3] What is the content of A[3] of the resulting array?
4.2 [3] What is the content of A[6] of the resulting array?
4.3 [3] What is the content of A[10] of the resulting array?
4.4 [3] What is the content of A[13] of the resulting array?
4.5 [3] We delete the root node from the heap and make it into a max heap again. What is the content of the A[6] of the resulting array?

5. [15] Let one-node binary tree have a depth of 1. Suppose we have a complete binary tree *T* of depth 100.
5.1 [3] What is the minimum number of nodes in *T*?
5.2 [3] What is the maximum number of nodes in *T*?
5.3 [3] There are *n* possible complete binary trees of depth 100. What is *n*?
5.4 [3] Suppose we number the nodes *T* by starting with the node on level 1, and continuing with the nodes on level 2, and so on. Nodes on any level are numbered from left to right. Let the first node be numbered 1, the second node be numbered 2, and so on. For a node numbered 200, what is its left node numbered?
5.5 [3] Following above, for a node numbered 3001, what is its parent node numbered?

6. [10] Suppose we have the following 11 integers:
44, 30, 95, 33, 50, 82, 18, 55, 70, 64, 26
Please create a hash table AA with 13 entries declared by
int AA[13];
Let's insert the integers one by one and from the left to the right into the hash table. Assume the hash function is h(k) = k%13 and linear probing is used for collision resolution. Note that % yields the remainder when one integer is divided by another, for example, 20%13 = 7 and 5%13 = 5. Please answer the following questions:
6.1 [2] What is the content of A[0]?
6.2 [2] What is the content of A[3]?
6.3 [2] What is the content of A[6]?
6.4 [2] What is the content of A[9]?
6.5 [2] What is the content of A[12]?

7. [15] Suppose we have 12 integers: 60, 40, 25, 10, 50, 80, 15, 65, 30, 90, 20, 55. Construct a binary search tree by inserting the integers, starting from 60, one after another. Please answer the following questions for the binary search tree:
7.1 [3] What is the left child of the root node?
7.2 [3] What is the right child of the node containing 80?
7.3 [3] What is the left child of the node containing 25?
7.4 [3] What is the right child of the node containing 55?
7.5 [3] What is the height of the binary tree? Let one-node tree have a depth of 1.