※ 考生請注意：本試題不可使用計算機。　請於答案卷(卡)作答，於本試題紙上作答者，不予計分。

1. Analyzing the time and space complexities for the following function and showing the final answer. (10%)
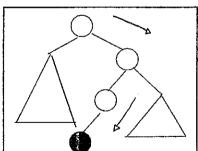
```
float sum(float list[ ], int n)
{
    float tempsum = 0;
    int i;
    for (i = 0; i < n; i++) {
        tempsum += list[i];
    }
    return tempsum;
```

2. Rewrite the function in Question 1 (question above) into recursive function and do the time and space complexity analysis again. The recursive function here has greater space complexity, why? (10%)

3. Please use one-dimension array to simulate a binary tree (10%)

4. The following program is to find the successor of a node for inorder, please modify the program to find the precedence of the node for inorder. (10%)

```
threaded_pointer insucc(threaded_pointer tree)
{
  threaded_pointer temp;
  temp = tree->right_child;
  if (!tree->right_thread)
    while (!temp->left_thread)
      temp = temp->left_child;
  return temp;
```


5. The time complexity of Quicksort may vary with a given sequence to be sorted. Describe the situations of its values in best, average, and worst cases and also write down the values. (10%)

6. Compete the following recursive code for pre-order traversal. (10%)

```
void preorder(tree_pointer ptr)
/* preorder tree traversal */
{
    if (ptr) {
        printf("%d", ptr->data);
```

7. For the unsorted sequence: 27, 6, 38, 2, 60, 12, 58, 16, 47, 18　(1) Please write down every sequence of each step while applying Quicksort (2) Show the Mergesort for it by binary tree. (20%)

8. For the following sequence, 27, 7, 66, 3, 55, 22, 81, 12, 1, 99, please step by step to complete the followings. (20%)

   (1) Create a MAX Heap using bottom-up approach
   (2) Show how you finish the Heapsort by the MAX heap you built on (1)
   (3) Analyze the time complexity for (1)+(2), that is the time complexity for Heapsort.