

1. (5 pt.)

10 is created  
 20 is created  
 21 is created  
 30 is created  
 31 is created  
 30 is destroyed  
 21 is destroyed  
 20 is destroyed  
 40 is created  
 41 is created  
 40 is destroyed  
 10 is destroyed

2. (5 pt.)

- Text section – executable code
- Data section – Global variable and Static variables
- Heap – space for dynamic memory allocation
- Stack section – store local variables when a function all

3. (10 pt.)

a)  $f(n) = O(g(n))$

- $O(g(n)) = \{f(n) \mid \text{there exist positive constant } c \text{ and } n_0 \text{ such that } 0 \leq f(n) \leq cg(n) \text{ for all } n \geq n_0\}$ .  $g(n)$  is upper bound of  $f(n)$

b)  $f(n) = \Theta(g(n))$

- $\Theta(g(n)) = \{f(n) \mid \text{there exist positive constant } c_1, c_2 \text{ and } n_0 \text{ such that } 0 \leq c_1g(n) \leq f(n) \leq c_2g(n) \text{ for all } n \geq n_0\}$ .  $g(n)$  is tight bound of  $f(n)$

c)  $f(n) = \Omega(g(n))$

- $\Omega(g(n)) = \{f(n) \mid \text{there exist positive constant } c \text{ and } n_0 \text{ such that } 0 \leq cg(n) \leq f(n) \text{ for all } n \geq n_0\}$ .  $g(n)$  is lower bound of  $f(n)$

4. (5 pt.)

Sol) By asymptotic notation, we need show that there exist constant  $C_1, C_2$  and  $n_0$  such that

$$C_1 n^2 \leq n^2 + 2n \leq C_2 n^2 \text{ for all } n_0 \geq 0$$

$$C_1 n^2 \leq n^2 + 2n, \text{ where } C_1 = 1 \text{ with } n_0 = 1$$

$$n^2 + 2n \leq C_2 n^2, \text{ where } C_2 = 2 \text{ with } n_0 = 2$$

$$C_1 n^2 \leq n^2 + 2n \leq C_2 n^2 \text{ with } C_1 = 1, C_2 = 2, n_0 = 2 \text{ true}$$

5. (5 pt.)

Sol) By asymptotic notation, there exist constant  $C$ , and  $n_0$  such that

$$\frac{1}{2}n^2 - 3n \leq Cn^2 \text{ with } C = 1 \text{ and } n_0 = 1, \text{ it is true}$$

6. (15 pt.)

a) Different	b) 4 6 46
c) 1 2 3 5 6 7 9 10	d) ood for you o
e) 1 2 3	f) 3 4 5 6
g) 4	h) GHIJK

7. (15 pt.)

```
bool Palindrome (char A[], int Size)
{
    int front =0;
    int back =Size -1;
    bool rval;
    while ((A[front]==A[back]) && (front <= back))
    {
        front++;
        back --;
    }
    if (front > back)
        rval = true;
    else
        rval = false;
    return rval;
}
```

8. (10 pt)

Insertion Sort	Selection Sort	Bubble Sort
6, 8, 7, 3, 2, 4, 5, 1	6, 8, 7, 3, 2, 4, 5, 1	6, 8, 7, 3, 2, 4, 5, 1
<b>6, 8</b> , 7, 3, 2, 4, 5, 1	<b>1</b> , 8, 7, 3, 2, 4, 5, 6	6, 7, 3, 2, 4, 5, 1, <b>8</b>
<b>6, 7, 8</b> , 3, 2, 4, 5, 1	<b>1, 2</b> , 7, 3, 8, 4, 5, 6	6, 3, 2, 4, 5, 1, <b>7, 8</b>
<b>3, 6, 7, 8</b> , 2, 4, 5, 1	<b>1, 2, 3</b> , 7, 8, 4, 5, 6	3, 2, 4, 5, 1, <b>6, 7, 8</b>
<b>2, 3, 6, 7, 8</b> , 4, 5, 1	<b>1, 2, 3, 4</b> , 8, 7, 5, 6	2, 3, 4, 1, <b>5, 6, 7, 8</b>
<b>2, 3, 4, 6, 7, 8</b> , 5, 1	<b>1, 2, 3, 4, 5</b> , 7, 8, 6	2, 3, 1, <b>4, 5, 6, 7, 8</b>
<b>2, 3, 4, 5, 6, 7, 8</b> , 1	<b>1, 2, 3, 4, 5, 6</b> , 8, 7	2, 1, <b>3, 4, 5, 6, 7, 8</b>
<b>1, 2, 3, 4, 5, 6, 7, 8</b>	<b>1, 2, 3, 4, 5, 6, 7, 8</b>	1, <b>2, 3, 4, 5, 6, 7, 8</b>

9. (15 pt.)

```

void selectionSort (int A[], int size)
{
    int min, tmp;
    for (int j = 0; j < size - 1; j++)
    {
        min = j;
        for (int k = j + 1; k <= size - 1; k++)
        {
            if (A[k] < A[min])
                min = k;
        }
        if (min != j)
        {
            tmp = A[j];
            A[j] = A[min];
            A[min] = tmp;
        }
    }
}

```

10. (15 pt.)

```
Node *DeleteNode (Node *List, int ID)
{
    Node *Tmp = List;

    // Case 1: List is empty
    if (List == NULL)
        cout << "Empty List" <<endl;
    //Case 2: Delete the first node
    else if (Tmp->IDNumber == ID)
    {
        List = List->Next;
        delete Tmp;
    }
    else
    {
        while (Tmp->Next !=NULL && Tmp->Next->ID != ID)
            Tmp = Tmp->Next;
        // Case 3: No Such a node
        if (Tmp ->Next == NULL)
            cout <<"There is no student with ID number: "<<ID <<endl;
        // Case 4: Delete a node from a list (not first node)
        else if (Tmp ->Next->IDNumber == ID)
        {
            Node *T = Tmp->Next;
            Tmp->Next = Tmp->Next->Next;
            delete T;
        }
    }
    return List;
}
```