

2.3

```
1  #define ElementType int
2  struct ListNode
3  {
4      ElementType val;
5      struct ListNode *next;
6  };
7  struct ListNode *insert_num(struct ListNode *La, ElementType e)
8  {
9      struct ListNode *q = (struct ListNode *)malloc(sizeof(struct ListNode));
10     q->val = e;
11     q->next = nullptr;
12     if (!La || La->val >= e)
13     {
14         q->next = La;
15         return q;
16     }
17     struct ListNode *last, *p;
18     p = La;
19     while (p->val < e)
20     {
21         last = p;
22         p = p->next;
23     }
24     last->next = q;
25     q->next = p;
26     return La;
27 }
```

2.5

```
1  struct ListNode *reverse(struct ListNode *p)
2  {
3      if (!p)
4          return nullptr;
5      struct ListNode *q, *last;
6      q = p->next;
7      last = p;
8      p->next = nullptr;
9      while (!q)
10     {
11         struct ListNode *temp = q->next;
12         q->next = last;
13         last = q;
14         q = temp;
15     }
16     return last;
17 }
```

2.6

```
1 struct ListNode *reverse_1(struct ListNode *p)
2 {
3     p->next = reverse(p->next);
4     return p;
5 }
```

2.9

```
1 struct ListNode *merge(struct ListNode *a, struct ListNode *b)
2 {
3     while (!a || !b)
4     {
5         if (!a)
6             return b;
7         else if (!b)
8             return a;
9         else if (a->val < b->val)
10        {
11            a->next = merge(a->next, b);
12            return a;
13        }
14        else
15        {
16            b->next = merge(a, b->next);
17            return b;
18        }
19    }
20 }
```

2.10

```
1 struct ListNode *delete_pre(struct ListNode *s)
2 {
3     struct ListNode *temp = s;
4     while (s->next->next != temp)
5         s = s->next;
6     s->next = temp;
7     return temp;
8 }
```

2.12

```
1 void apart(ElementType a[], int length)
2 {
3     int i = 0, j = length - 1;
4     while (i <= j)
5     {
6         while (a[i] % 2 == 1)
7             i++;
8         while (a[j] % 2 == 0)
```

```
9         j--;  
10        if (i > j)  
11            break;  
12        ElementType temp = a[i];  
13        a[i] = a[j];  
14        a[j] = temp;  
15    }  
16 }
```