数据结构作业 2.3/2.6/2.9/2.12/2.15/2.1…

- 姓名：牟鑫一
- 班级：191174班

2.3

2.6 线性表原地逆置

```
template <class T>
bool Inverse(T A[], int arraySize, int n)
{
    if (n > arraySize) return false;
    T temp;
    for (int i = 0; i <= n / 2; i++)
    {
        temp = A[i];
        A[i] = A[n - i];
        A[n - i] = temp;
    }
}
```

2.9 将数组中两个顺序表的位置互换

```
template <class T>
bool Exchange(T A[], int m, int n)
{
    int temp;
    for (int i = 0; i <= (m + n) / 2; i++)
    {
        temp = A[i];
        A[i] = A[(m + n - 1) - i];
        A[(m + n - 1) - i] = temp;
    }
    for (int j = 0; j <= n / 2; j++)
    {
        temp = A[j];
        A[j] = A[(n - 1) - j];
        A[(n - 1) - j] = temp;
    }
    for (int k = 0; k <= m / 2; k++)
    {
        temp = A[n + k];
        A[n + k] = A[(m + n - 1) - k];
        A[(m + n - 1) - k] = temp;
    }
}
```

2.12 将两个顺序表A、B合并成一个顺序表C

```
template <class T>
bool Exchange(T A[], T B[], int m, int n)
{
     T C[m + n];
        C[i] = A[i];
    for (int j = 0; j < n; j++)
          C[m + j] - B[j];
    T temp;
    for (int k = 0; k < m + n; k++)
        for (int l = k+1; l < m + n; l++)
            if(C[k]>C[l])
            {
                temp = C[k];
                C[k] = C[l];
                C[l] = temp;
            }
}
```

2.15 合并两个两个非递减有序单链表成一个非递增有序单链表

```
template<class T>
void List<T>::Combine(List<T>& A, List<T>& B, T ha, T hb) {
    List<T> *current, *pre, *p, *q;
    LinkNode<A>* last = hb;
    p = ha->link;   ha->link = NULL;
    while (p!=NULL){
        current = ha->link; pre = ha;
        while (current != NULL && current->data <= p->data) {
            pre = current;current = current->link;
        }
        q = p;  p = p->link;    q->link = pre->link; pre->link = q;
    }
}
```

2.19 改造一个带附加头结点的双向链表，保持所有结点的原有次序在各个结点的rLink域中，并利用lLink域把所有结点按照其值从小到大的顺序连接起来

```
template< class T, class E>
void SortDbl(DblList< T, E> & DL) {
   DblNode<T, E> * pr, *p, *s, *h;
      h = DL.First();
      s = h->rLink->rLink;
      h->rLink->lLink = h;
      h->lLink = h->rLink;
      while (s != h) {
             pr = h;
             p = h->lLink;
         }
   while (p != h && p->data < s->data) {
                pr = p;
                p = p->lLink;
         }
         pr->lLink = s;
         s->lLink = p;
         s = s->rLink;
     }
}
```

2.22 删除带表头节点的数据值按递增顺序排列的单链表中所有大于min，小于max的元素

```
template<class T, class E>
void rangeDelete(List<T, E>& L, T min, T max) {
    LinkNode<T, E> *pr = L.First(), *p = pr->link;
    while (p != NULL && p->data <= min){
        pr = p; p = p->link;
    }
    while (p != NULL && p->data < max){
        pr->link = p->link;
        delete p;
        p = pr->link;
    }
}
```