

```

/* 归并排序 - 循环实现 */
/* 这里Merge函数在递归版本中给出 */

/* length = 当前有序子列的长度*/
void Merge_pass( ElementType A[], ElementType TmpA[], int N, int length )
{ /* 两两归并相邻有序子列 */
    int i, j;

    for ( i=0; i <= N-2*length; i += 2*length )
        Merge( A, TmpA, i, i+length, i+2*length-1 );
    if ( i+length < N ) /* 归并最后2个子列*/
        Merge( A, TmpA, i, i+length, N-1);
    else /* 最后只剩1个子列*/
        for ( j = i; j < N; j++ ) TmpA[j] = A[j];
}

void Merge_Sort( ElementType A[], int N )
{
    int length;
    ElementType *TmpA;

    length = 1; /* 初始化子序列长度*/
    TmpA = malloc( N * sizeof( ElementType ) );
    if ( TmpA != NULL ) {
        while( length < N ) {
            Merge_pass( A, TmpA, N, length );
            length *= 2;
            Merge_pass( TmpA, A, N, length );
            length *= 2;
        }
        free( TmpA );
    }
    else printf( "空间不足" );
}

```