

```
#include <iostream>
using namespace std ;

// -----
int * merge( int list0[], int n0, int list1[], int n1 )
{
    int * r ;
    int i0, i1, ir ;

    r = new int [ n0 + n1 ] ;
    i0 = 0 ; i1 = 0 ; ir = 0 ;

    while ( ( i0 < n0 ) && ( i1 < n1 ) ) {
        if ( list0[i0] < list1[i1] ) {
            r[ir] = list0[i0] ;
            i0++ ; ir++ ;
        }
        else {
            r[ir] = list1[i1] ;
            i1++ ; ir++ ;
        }
    }

    // One of the lists has run out. Copy remaining elements.

    while ( i0 < n0 ) { // If list0 is finished, loop is not entered.
        r[ir] = list0[i0] ;
        i0++ ; ir++ ;
    }

    while ( i1 < n1 ) { // If list1 is finished, loop is not entered.
        r[ir] = list1[i1] ;
        i1++ ; ir++ ;
    }

    return(r) ;
}

// -----
// Split the list in half. Recursively sort both halves, and merge the
// partial results back together. This approach to solving a problem
// is often called "divide and conquer".
// -----
int * mergesort( int * a, const int n )
{
    int n0, n1 ;
    int * r0, * r1, *r ;

    if ( n == 1 ) {
        r = new int [1] ;
        r[0] = a[0] ;
        return r ;
    }
    else {
        n0 = n / 2 ;
        n1 = n - n0 ;
        r0 = mergesort( a, n0 ) ;
        r1 = mergesort( (a+n0), n1 ) ;
        r = merge( r0, n0, r1, n1 ) ;
        delete [] r0 ; delete [] r1 ;
        return r ;
    }
}
}
```

```
// -----  
int main()  
{  
    int i ;  
    const int n = 12 ;  
    int * sort_a ;  
    int a[n] = { 17, 28, 3, 17, 12, 5, 9, 2, 23, 11, 7, 8 } ;  
  
    sort_a = mergesort( a, n ) ;  
  
    cout << endl ;  
    cout << "a = [ " ;  
    for ( i = 0 ; i < n ; i++ ) {  
        cout << a[i] ;  
        if ( (i+1) < n ) cout << ", " ;  
    }  
    cout << " ]" << endl << endl ;  
  
    cout << "sort_a = [ " ;  
    for ( i = 0 ; i < n ; i++ ) {  
        cout << sort_a[i] ;  
        if ( (i+1) < n ) cout << ", " ;  
    }  
    cout << " ]" << endl << endl ;  
}
```

```
----- Sample session -----  
atlas%  
atlas% g++ mergesort.cc  
atlas%  
atlas% a.out  
  
a = [ 17, 28, 3, 17, 12, 5, 9, 2, 23, 11, 7, 8 ]  
  
sort_a = [ 2, 3, 5, 7, 8, 9, 11, 12, 17, 17, 23, 28 ]
```