Worksheet W.A.24.1 Name________________________

Non-Recursive
Merge and Mergesort

1.

```
void mergeSort (int Numbers[], int first, int last)
{
   int mid = (first + last) / 2;
   selectionSort (Numbers, first, mid);
   selectionSort (Numbers, mid+1, last);
   merge (Numbers, first, mid, last);
}
```

Use this non-recursive *mergeSort* method to analyze the changes made to an array of integers - referred to as *Numbers* by this method. *Numbers* contains the following integers: 56 32 24 74 29 84 13 65 70 52 16 44 37. The parameter *first* receives the value of 0 since it is the index of the first integer of the array: 56. The parameter last receives the value of 12 since it is the index of the last integer of the array: 37.

Explain the result of each of the following lines of this method <u>and</u> the specific impact on the array *Numbers:*

```
int mid = (first + last) / 2;
```

```
selectionSort (Numbers, first, mid);
```

```
selectionSort (Numbers, mid+1, last);
```

```
merge (Numbers, first, mid, last);
```

2. The last line of the *Mergesort* method (above) calls the *merge* method that is discussed in sections B. 4, B.5 and B.6 in the Student Outline for this lesson. Assume that a new array of integers has had each of its two halves already sorted (using the calls to the selectionSort methods) and looks as follows. Rewrite the integers using the *merge* method on the next line with arrows to reflect the order of their movement.

12 23 28 31 47 65 3 15 18 35 44 59

— — — — — — — — — — — —