```c
#include <cs50.h>
#include <stdio.h>

void merge(int a[], int start, int mid, int mid_1, int end);
void merge_sort(int a[], int start, int end);

int main (void)
{
    // take in array size and elements from user
    printf("please enter array size: ");
    int n = get_int("");
    int array[n];
    printf("please enter %i elements, to fill array: \n", n);
    for(int i = 0; i < n; i++)
    {
        array[i] = get_int("");
    }

    // print unsorted array
    printf("unsorted array is: \n");
    for(int i = 0; i < n; i++)
    {
        printf("%i ", array[i]);
    }
    printf("\n");

    // run merge sort, and print sorted array
    merge_sort(array, 0, n - 1);
    printf("sorted array is: \n");
    for(int i = 0; i < n; i++)
    {
        printf("%i ", array[i]);
    }
    printf("\n");
}

// incomplete implementation of merge sort
void merge_sort(int a[], int start, int end)
{
    if (end > start)
    {
        int middle = (start + end) / 2;
```

```
43          merge_sort(a, start, middle);
44          merge_sort(a, middle + 1, end);
45          merge(a, start, middle, middle + 1, end);
46      }
47  }
48
49  void merge(int a[], int start, int mid, int mid_1, int end)
50  {
51      // declare temp array
52      int b[end - start + 1];
53
54      // iterate through both halves, placing whichever is the higher value into temp array
55      int i = 0, j = 0;
56      while (start + i <= mid && mid_1 + j <= end)
57      {
58          if (a[start + i] < a[mid_1 + j])
59          {
60              b[i + j] = a[start + i];
61              i++;
62          }
63          else
64          {
65              b[i + j] = a[mid_1 + j];
66              j++;
67          }
68      }
69
70      // copy elements from either half that remain, after direct comparison
71      while (start + i <= mid)
72      {
73          b[i + j] = a[start + i];
74          i++;
75      }
76      while (mid_1 + j <= end)
77      {
78          b[i + j] = a[mid_1 + j];
79          j++;
80      }
81
82      // copy temp array into final array
83      for (int k = 0; k < i + j; k++)
84      {
```

```
85          a[start + k] = b[k];
86      }
87  }
```