Question 1: You are sorting a 17-element vector, **v** using merge sort. The prototype for your recursive merge sort procedure is:

```
void recursive_sort(vector <double> &v, vector <double> &temp, int start, int size);
```

The initial call to `recursive_sort()` will be:

```
recursive_sort(v, temp, 0, 17);
```

What two calls to **recursive_sort()** will this initial call make?

- **A**: `recursive_sort(v, temp, 0, 17)` and `recursive_sort(v, temp, 8, 17)`.
- **B**: `recursive_sort(v, temp, 0, 8)` and `recursive_sort(v, temp, 8, 8)`.
- **C**: `recursive_sort(v, temp, 0, 8)` and `recursive_sort(v, temp, 8, 9)`.
- **D**: `recursive_sort(v, temp, 0, 8)` and `recursive_sort(v, temp, 9, 8)`.
- **E**: `recursive_sort(v, temp, 0, 8)` and `recursive_sort(v+8, temp, 0, 9)`.

Question 2: Suppose you are sorting the following vector with Quicksort, using median-of-three.

```
  21   25   44   05   30   07   82   10   28
```

What will be the pivot?

Questions 3 and 4: Suppose you are sorgint the string "SUFOJVEA" using merge sort. Right before you perform the final merge, what are the two substrings that you are merging? Enter the first substring as the answer to question 3, and the second as the answer to question 4.

Answers to Clicker Questions

Question 1: You divide size in half: 17/2 = 8. Your first recursive call works on the first "half" of the elements:

```
recursive_sort(v, temp, 0, 8)
```

Your second recursive call works on the remaining elements:

```
recursive_sort(s, temp, 8, 9).
```

The answer is **C**.

Question 2: It is the median of the first element (21), the last element (28) and the middle element (30). The answer is 28.

Questions 3 and 4: You recursively sort the first half of the string: "SUFO" becomes "FOSU". Then you recursively sort the second half of the string: "JVEA" becomes "AEJV". Those are the two answers:

Question 3: FOSU
Question 4: AEJV