Sort a given set of N integers elements using quick sort technique and compute its time taken

```
#include <stdio.h>
#include <time.h>

void quicksort (int number [25], int first, int last)
{
  int i, j, pivot, temp;
  if (first < last)
  {
    pivot = first;
    i = first;
    j = last;
    while (i < j)
    {
      while (number [i] <= number [pivot] &&
             i < last)
        i++;
      while (number [j] > number [pivot])
        j--;
      if (i < j)
      {
        temp = number [i];
        number [i] = number [j];
        number [j] = temp;
      }
    }
```

```
        temp = numbers[pivot];
        numbers[pivot] = numbers[j];
        numbers[j] = temp;
    quicksort(numbers, first, j-1);
    quicksort(numbers, j+1, last);
    }
}

int main()
{
    int i, count, numbers[25];
    clock_t start, end;
    start = clock();
    printf("enter elements: ");
    scanf("%d", &count);
    printf("\nEnter %d elements.", count);
    for(i=0; i<count; i++)
    scanf("%d", &numbers[i]);
    quicksort(numbers, 0, count-1);
    end = clock();
    printf("%d", numbers[i]);
printf("\n Time taken to sort is: %f", difftime
        difftime(end, start)/(clocks_per_sec));
    return 0;
}
```

Output

```
enter elements: 5
enter 5 elements: 30 10 80 90 40
sorted elements: 10 30 40 80 90
```

```
Enter the number of elements: 7
Enter the elements:
88
-5
65
-10
0
25
18
Original array: 88 -5 65 -10 0 25 18
Sorted array: -10 -5 0 18 25 65 88

Process returned 0 (0x0)   execution time : 27.711 s
Press any key to continue.
```