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

// Function to swap two elements  void
swap(int* a, int* b) {
     int temp = *a;
   *a = *b;
   *b = temp;
}

// Function to partition the array and return the pivot index int
partition(int arr[], int low, int high) {
   int pivot = arr[high];
    int i = (low - 1);

   for (int j = low; j <= high - 1; j++)
   { if (arr[j] < pivot) {
         i++;
         swap(&arr[i], &arr[j]);
      }
   }
   swap(&arr[i + 1], &arr[high]);
    return (i + 1);
}

// Function to perform Quick Sort
void quickSort(int arr[], int low, int high)
 {  if (low < high) {
      int pi = partition(arr, low, high);

      quickSort(arr, low, pi - 1);
      quickSort(arr, pi + 1, high);
   }
```

```
}

// Function to generate random numbers between 0 and 999 int
generateRandomNumber() {
    return rand() % 1000;
}

int main() {
    // Set n value int n
    = 6000;

    // Allocate memory for the array
    int* arr = (int*)malloc(n * sizeof(int));

    // Generate random elements for the array
    srand(time(NULL));
    printf("Random numbers for n = %d:\n", n); for
    (int i = 0; i < n; i++) {
        arr[i] = generateRandomNumber();
        printf("%d ", arr[i]);
    }
    printf("\n");

    // Record the start time
    clock_t start = clock();

    // Perform quick sort
    quickSort(arr, 0, n - 1);

    // Record the end time
    clock_t end = clock();

    // Calculate the time taken for sorting
    double time_taken = ((double)(end - start)) / CLOCKS_PER_SEC;

    // Output the time taken to sort for the current value of n
    printf("\nTime taken to sort for n = %d: %lf seconds\n\n", n, time_taken);
    // Display sorted numbers
    printf("Sorted numbers for n = %d:\n", n);  for
    (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n\n");

    // Free the dynamically allocated memory  free(arr);

    return 0;
}
```