```
#include  <stdio.h>
#include <stdlib.h>
#include <time.h>
// Function to perform selection sort
void selectionSort(int arr[], int n)
 {
    int i, j,  minIndex,  temp;
     for (i = 0; i < n - 1; i++) {
        minIndex = i;
            for (j = i + 1; j < n; j++) {
             if (arr[j] < arr[minIndex])
             {  minIndex = j;
             }
        }
        // Swap the found minimum element with the first element  temp
        = arr[minIndex];
        arr[minIndex] = arr[i];
        arr[i] = temp;
    }
}
// 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 selection sort
    selection Sort(arr, n);
    // 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;
}
```