Description:

C program that sorts an integer array using insertion sort ($O(n^2)$), prints the array before and after sorting, and reports the CPU time taken using clock().

Source Code:

```
#include <stdio.h>

#include <time.h>


void insertionSort(int array[], int n) {

	int i, element, j;


	for (i = 1; i < n; i++) {

	element = array[i];

	j = i - 1;


	// Move elements of array[0..i-1], that are greater than element,

	// to one position ahead of their current position

	while (j >= 0 && array[j] > element) {

	array[j + 1] = array[j];

	j = j - 1;

	}

	array[j + 1] = element;

	}

}


void printArray(int array[], int n) {
```

```
    int i;
    for (i = 0; i < n; i++) {
    printf("%d ", array[i]);
    }
    printf("\n");
}

int main() {
    int arr[] = {50, 23, 9, 18, 61, 32};

    clock_t start, end;
    double cpu_time_used;

    // Calculate the number of elements in the array
    int n = sizeof(arr) / sizeof(arr[0]);

    printf("\nBefore sorting: ");
    printArray(arr, n);

    start = clock();

    insertionSort(arr, n);
    end = clock();
    printf("\nAfter sorting: ");
    printArray(arr, n);
    cpu_time_used = ((double)(end - start)) / CLOCKS_PER_SEC;
```

```
    printf("Execution time: %.6f seconds\n", cpu_time_used);

    return 0;
}
```

Output:

```
Before sorting: 50 23 9 18 61 32
After sorting: 9 18 23 32 50 61
Execution time: 0.000002 seconds
```