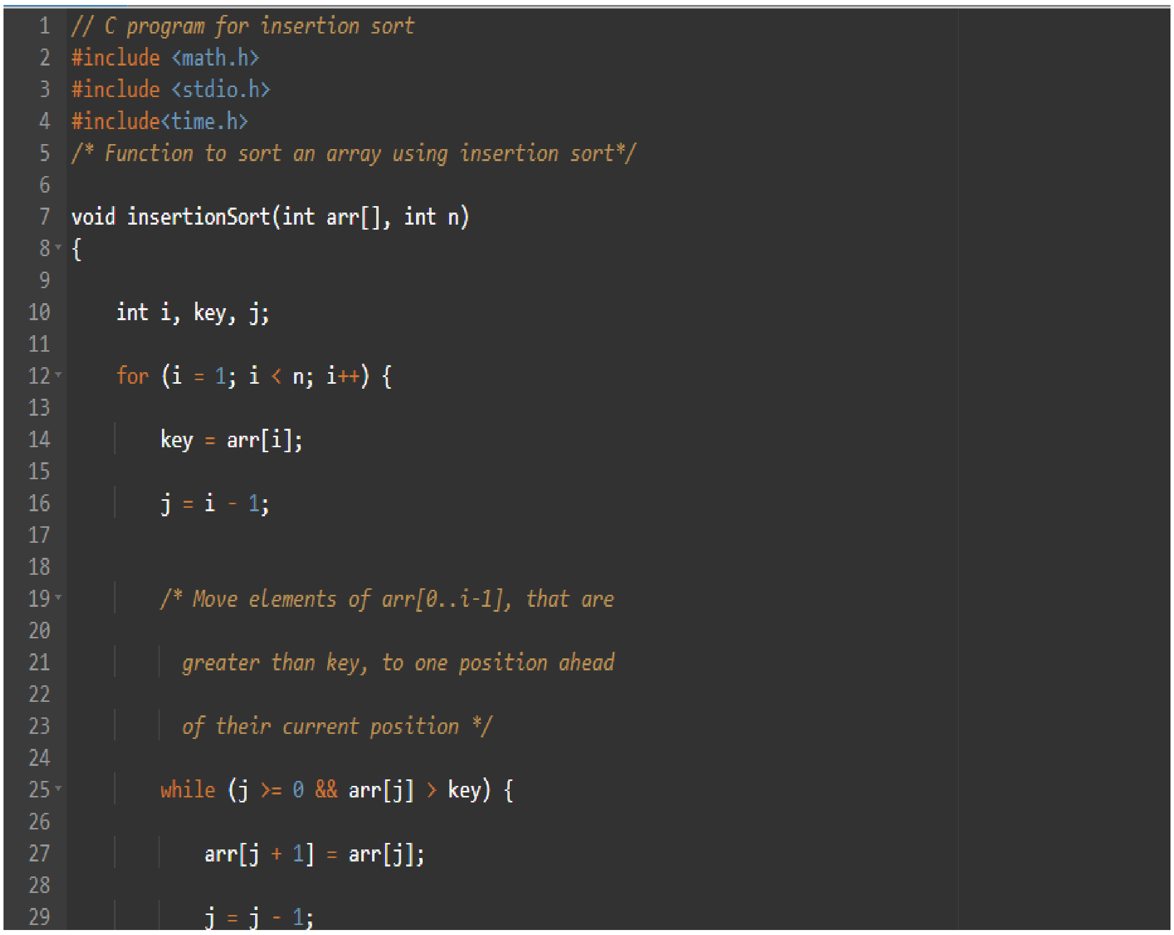
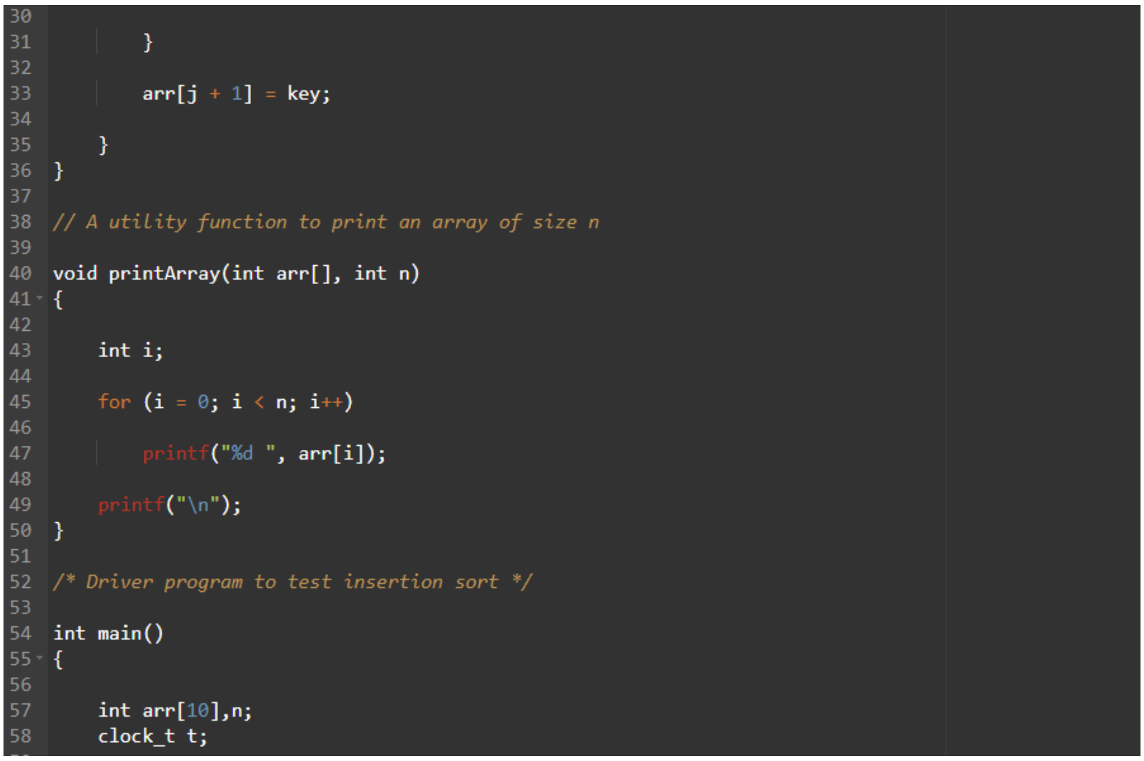
Output:

<u>Code:</u>

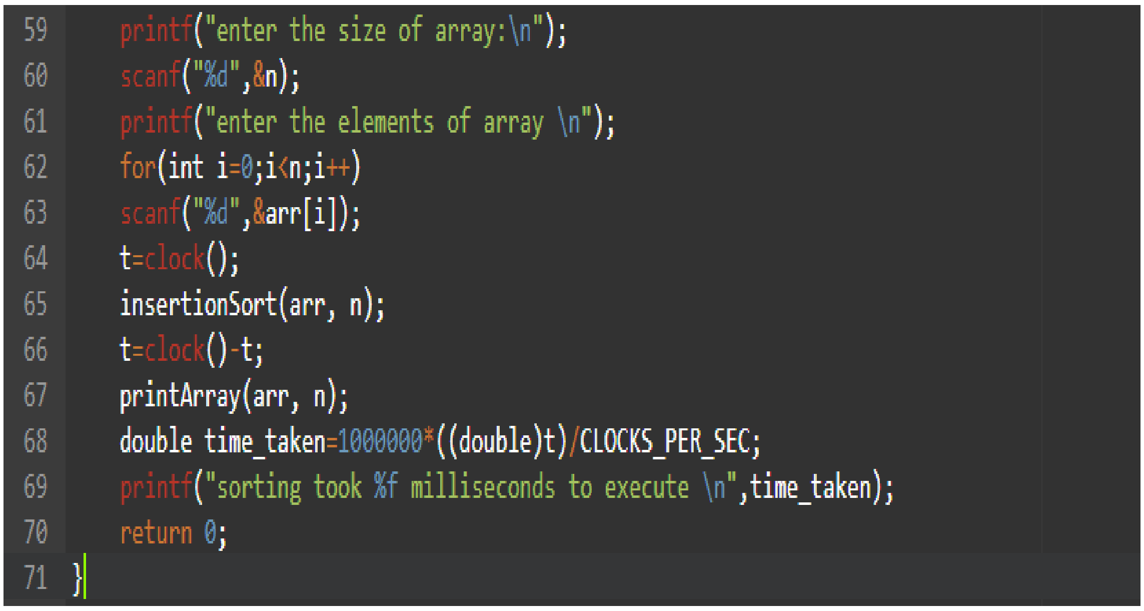


```c
// C program for insertion sort
#include <math.h>
#include <stdio.h>
#include<time.h>
/* Function to sort an array using insertion sort*/

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

    int i, key, j;

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

        key = arr[i];

        j = i - 1;


        /* Move elements of arr[0..i-1], that are

          greater than key, to one position ahead

          of their current position */

        while (j >= 0 && arr[j] > key) {

            arr[j + 1] = arr[j];

            j = j - 1;
```

Output:

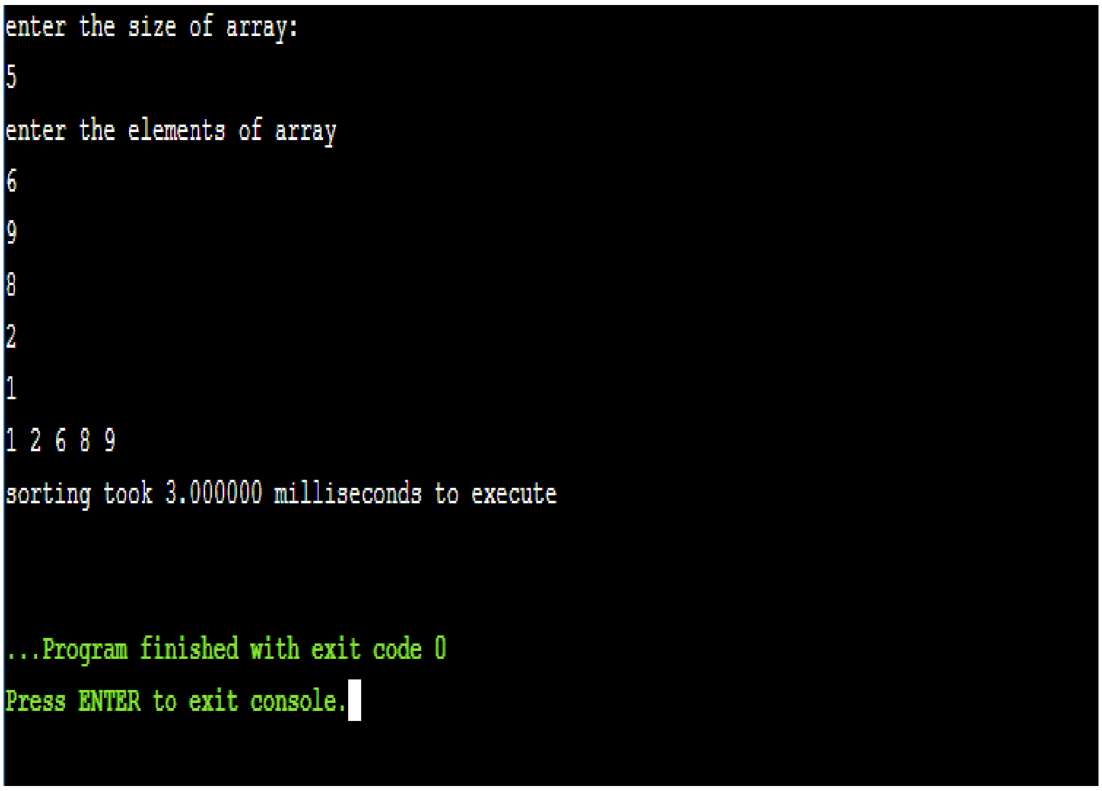


```c

        }

        arr[j + 1] = key;

    }
}

// A utility function to print an array of size n

void printArray(int arr[], int n)
{

    int i;

    for (i = 0; i < n; i++)

        printf("%d ", arr[i]);

    printf("\n");
}

/* Driver program to test insertion sort */

int main()
{

    int arr[10],n;
    clock_t t;
```



```c
    printf("enter the size of array:\n");
    scanf("%d",&n);
    printf("enter the elements of array \n");
    for(int i=0;i<n;i++)
    scanf("%d",&arr[i]);
    t=clock();
    insertionSort(arr, n);
    t=clock()-t;
    printArray(arr, n);
    double time_taken=1000000*((double)t)/CLOCKS_PER_SEC;
    printf("sorting took %f milliseconds to execute \n",time_taken);
    return 0;
}
```

Output:



```
enter the size of array:
5
enter the elements of array
6
9
8
2
1
1 2 6 8 9
sorting took 3.000000 milliseconds to execute


...Program finished with exit code 0
Press ENTER to exit console.
```