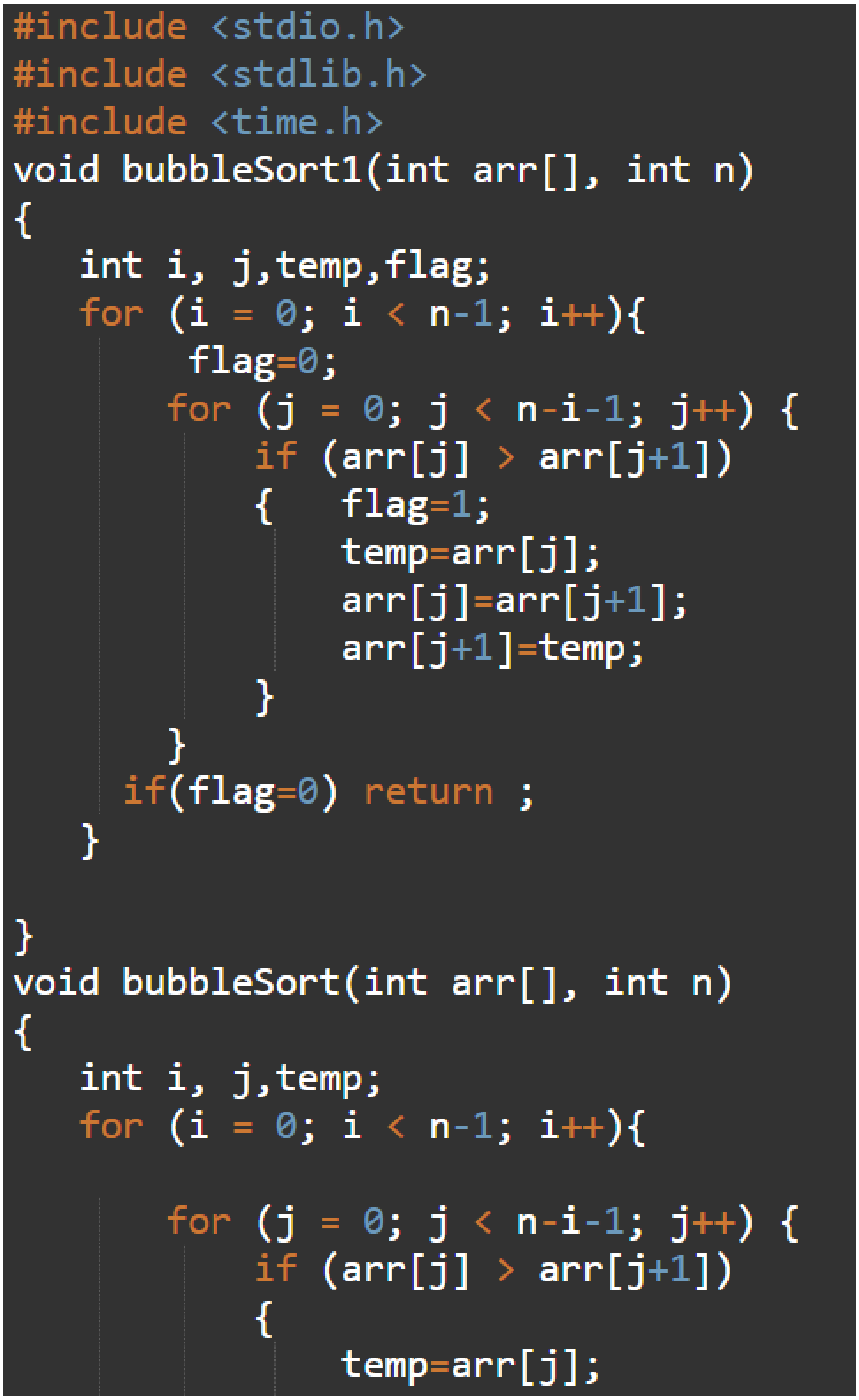
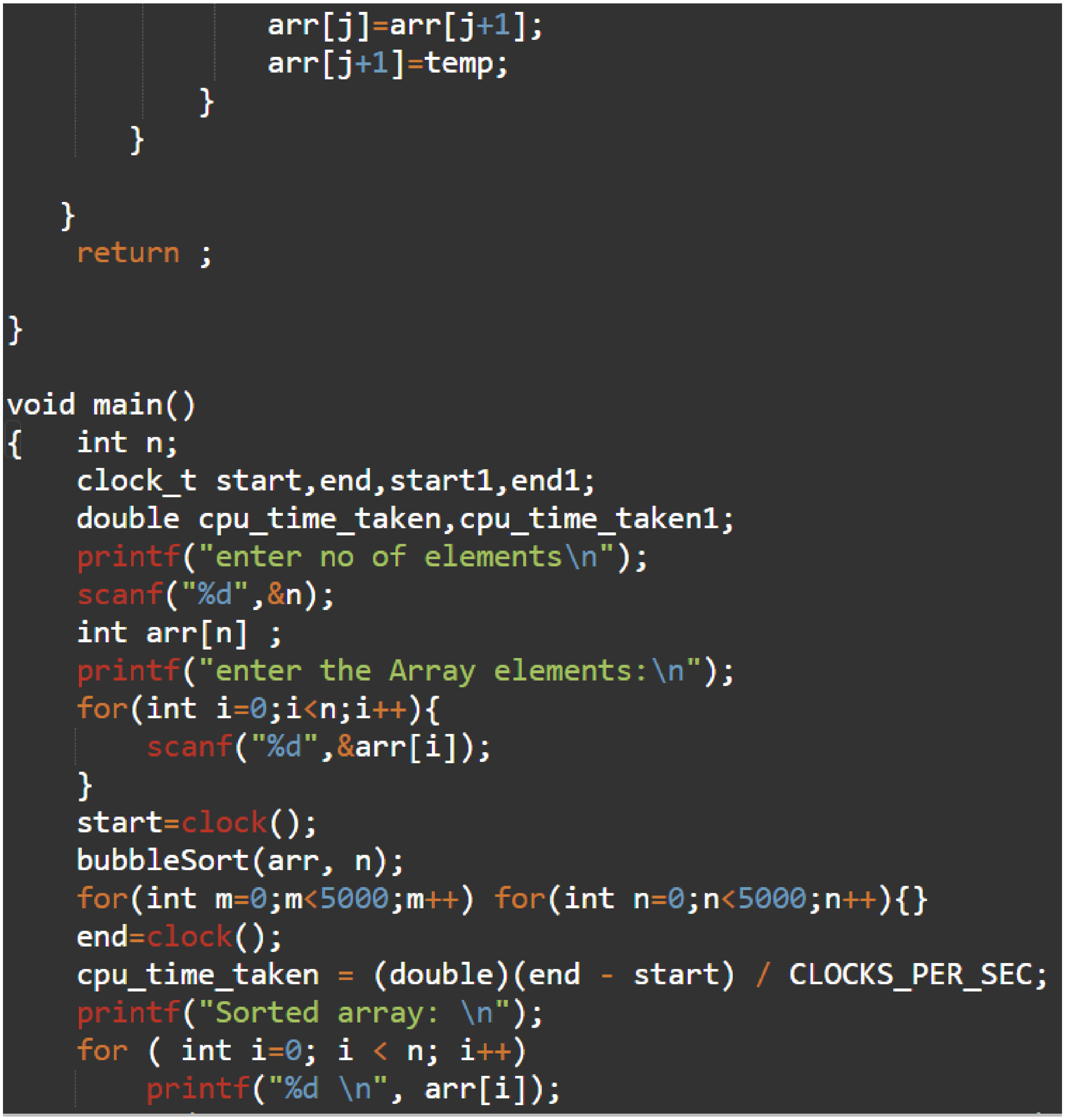
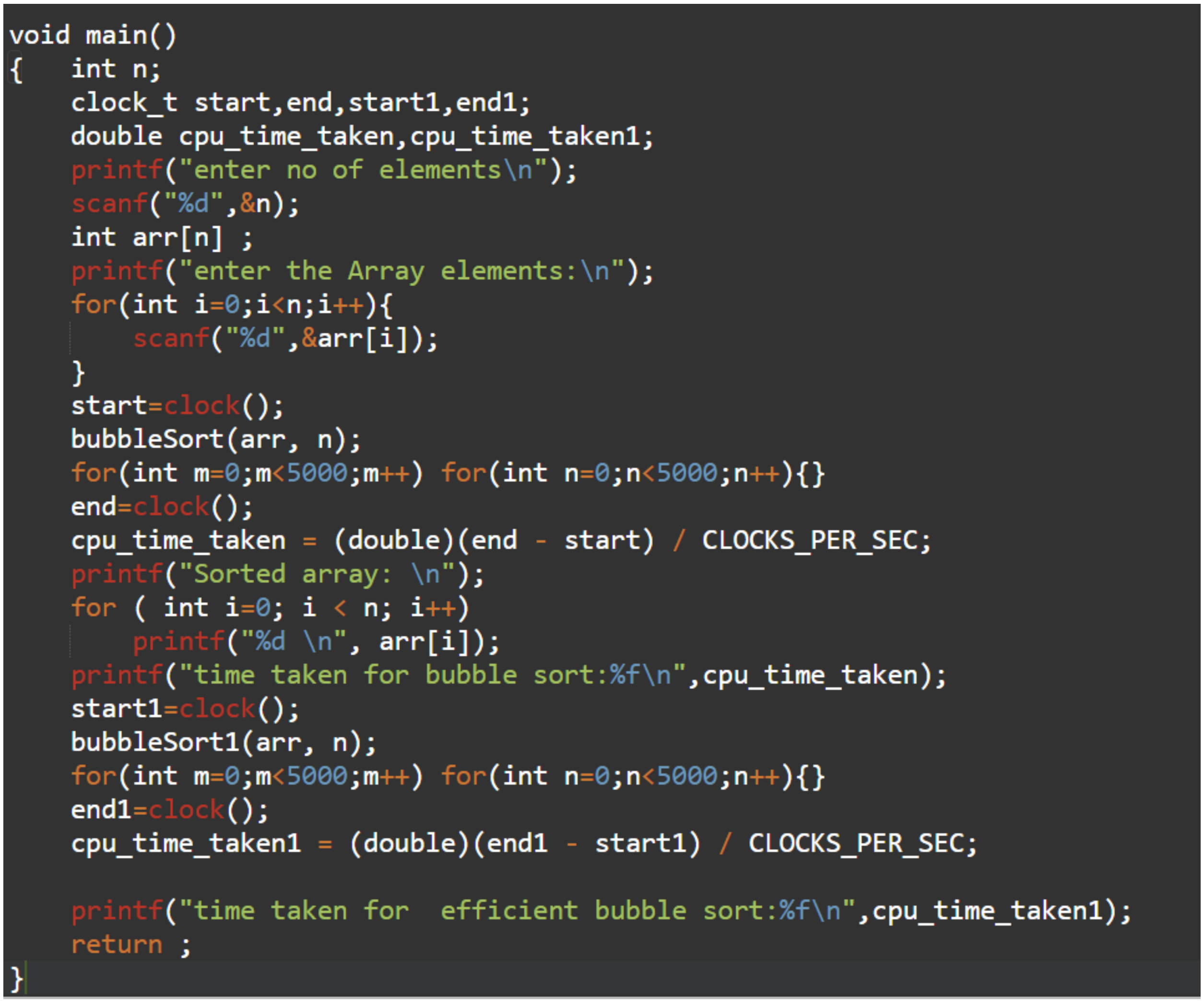
Bubble sort program
program :-



```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
void bubbleSort1(int arr[], int n)
{
   int i, j,temp,flag;
   for (i = 0; i < n-1; i++){
        flag=0;
       for (j = 0; j < n-i-1; j++) {
           if (arr[j] > arr[j+1])
           {   flag=1;
               temp=arr[j];
               arr[j]=arr[j+1];
               arr[j+1]=temp;
           }
       }
     if(flag=0) return ;
   }

}
void bubbleSort(int arr[], int n)
{
   int i, j,temp;
   for (i = 0; i < n-1; i++){

       for (j = 0; j < n-i-1; j++) {
           if (arr[j] > arr[j+1])
           {
               temp=arr[j];
```

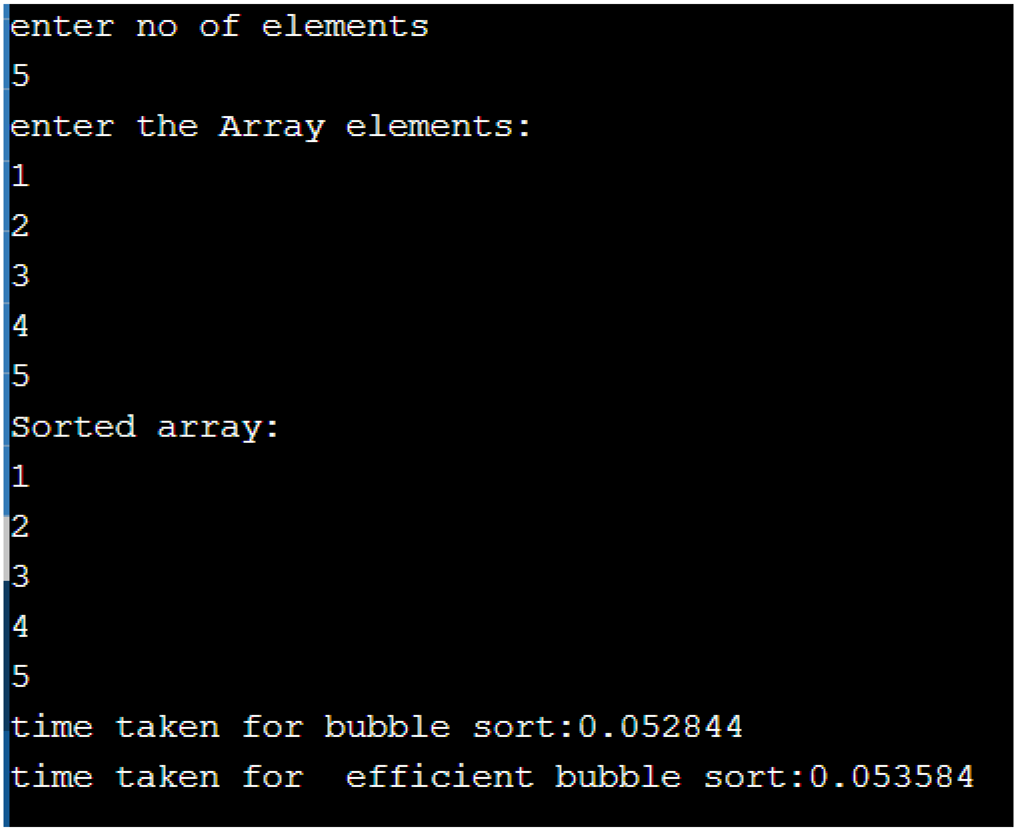


```
               arr[j]=arr[j+1];
               arr[j+1]=temp;
           }
       }

   }
    return ;

}

void main()
{   int n;
    clock_t start,end,start1,end1;
    double cpu_time_taken,cpu_time_taken1;
    printf("enter no of elements\n");
    scanf("%d",&n);
    int arr[n] ;
    printf("enter the Array elements:\n");
    for(int i=0;i<n;i++){
        scanf("%d",&arr[i]);
    }
    start=clock();
    bubbleSort(arr, n);
    for(int m=0;m<5000;m++) for(int n=0;n<5000;n++){}
    end=clock();
    cpu_time_taken = (double)(end - start) / CLOCKS_PER_SEC;
    printf("Sorted array: \n");
    for ( int i=0; i < n; i++)
        printf("%d \n", arr[i]);
```



```
void main()
{   int n;
    clock_t start,end,start1,end1;
    double cpu_time_taken,cpu_time_taken1;
    printf("enter no of elements\n");
    scanf("%d",&n);
    int arr[n] ;
    printf("enter the Array elements:\n");
    for(int i=0;i<n;i++){
        scanf("%d",&arr[i]);
    }
    start=clock();
    bubbleSort(arr, n);
    for(int m=0;m<5000;m++) for(int n=0;n<5000;n++){}
    end=clock();
    cpu_time_taken = (double)(end - start) / CLOCKS_PER_SEC;
    printf("Sorted array: \n");
    for ( int i=0; i < n; i++)
        printf("%d \n", arr[i]);
    printf("time taken for bubble sort:%f\n",cpu_time_taken);
    start1=clock();
    bubbleSort1(arr, n);
    for(int m=0;m<5000;m++) for(int n=0;n<5000;n++){}
    end1=clock();
    cpu_time_taken1 = (double)(end1 - start1) / CLOCKS_PER_SEC;

    printf("time taken for  efficient bubble sort:%f\n",cpu_time_taken1);
    return ;
}
```

OUTPUT:-



```
enter no of elements
5
enter the Array elements:
1
2
3
4
5
Sorted array:
1
2
3
4
5
time taken for bubble sort:0.052844
time taken for  efficient bubble sort:0.053584
```