

```
main.c
8
9 #include<stdio.h>
10 #include<time.h>
11 #include<stdlib.h>
12 void median(int arr[],int size){
13     float n=(arr[size/2]+arr[(size-1)/2])/2.0;
14     printf("\nThe median of the array is: %f",n);
15 }
16 void sort(int arr[],int size){
17     int temp,i,j;
18     for(i=0;i<size;i++){
19         for(j=i;j<size;j++){
20             if(arr[j]<arr[i]){
21                 temp=arr[i];
22                 arr[i]=arr[j];
23                 arr[j]=temp;
24             }
25         }
26     }
27 }
28 void printArray(int arr[],int size){
29     for(int i=0;i<size;i++){
30         printf("%d\n",arr[i]);
31     }
32 }
33 int main(){
34     int n,i,j;
35     printf("Enter size of both arrays: ");
36     scanf("%d",&n);
```

```
main.c
34 int n,i,j;
35 printf("Enter size of both arrays: ");
36 scanf("%d",&n);
37 int a[n],b[n],c[2*n];
38 for(i=0;i<n;i++){
39     a[i]=rand();
40 }
41 for(i=0;i<n;i++){
42     b[i]=rand();
43 }
44 printf("\nElements of first array: \n");
45 printArray(a,n);
46 printf("\nElements of second array: \n");
47 printArray(b,n);
48 sort(a,n);
49 sort(b,n);
50 printf("\nSorted first array: \n");
51 printArray(a,n);
52 printf("\nSorted second array: \n");
53 printArray(b,n);
54 clock_t begin = clock();
55 for(j=0;j<n;j++){
56     c[j]=a[j];
57 }
58 for(int k=0;k<n;k++){
59     c[j+k]=b[k];
60 }
61 printf("\nMerged array is: \n");
62 printArray(c,2*n);
63 median(c,2*n);
```



```
main.c
39     a[i]=rand();
40 }
41 for(i=0;i<n;i++){
42     b[i]=rand();
43 }
44 printf("\nElements of first array: \n");
45 printArray(a,n);
46 printf("\nElements of second array: \n");
47 printArray(b,n);
48 sort(a,n);
49 sort(b,n);
50 printf("\nSorted first array: \n");
51 printArray(a,n);
52 printf("\nSorted second array: \n");
53 printArray(b,n);
54 clock_t begin = clock();
55 for(j=0;j<n;j++){
56     c[j]=a[j];
57 }
58 for(int k=0;k<n;k++){
59     c[j+k]=b[k];
60 }
61 printf("\nMerged array is: \n");
62 printArray(c,2*n);
63 median(c,2*n);
64 clock_t end = clock();
65 double time_spent = (double)(end - begin) / CLOCKS_PER_SEC;
66 printf("\nExecution Time : %f seconds\n", time_spent);
67 return 0;
```

```
← → ↻ onlinegdb.com/online_c_compiler
input
Enter size of both arrays: 3

Elements of first array:
1804289383
846930886
1681692777

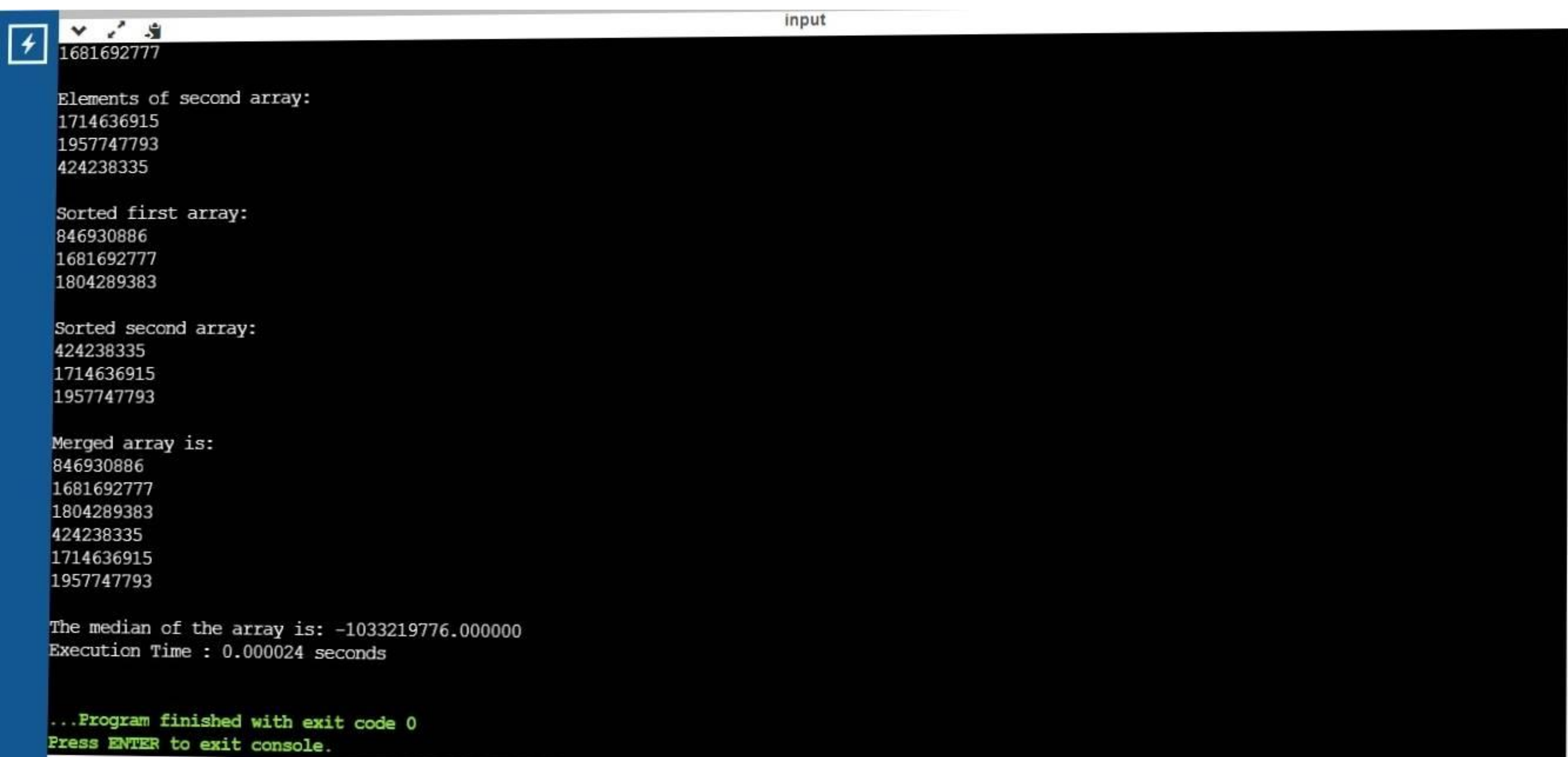
Elements of second array:
1714636915
1957747793
424238335

Sorted first array:
846930886
1681692777
1804289383

Sorted second array:
424238335
1714636915
1957747793

Merged array is:
846930886
1681692777
1804289383
424238335
1714636915
1957747793

The median of the array is: -1033219776.000000
```



```
1681692777
Elements of second array:
1714636915
1957747793
424238335

Sorted first array:
846930886
1681692777
1804289383

Sorted second array:
424238335
1714636915
1957747793

Merged array is:
846930886
1681692777
1804289383
424238335
1714636915
1957747793

The median of the array is: -1033219776.000000
Execution Time : 0.000024 seconds

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