Logs & others

Code::Blocks | Search results | Cccc | Build log | Build messages

Start here | merge sort.c | MEDIAN OF TWO SORTED ARRAY AFTER MERGING.c

```c
#include<stdio.h>
#include<time.h>
#include<stdlib.h>

void median(int arr[],int size)
{
    float n=(arr[size/2]+arr[(size-1)/2])/2.0;
    printf("\nThe median of the array is: %f",n);
}

void sort(int arr[],int size)
{
    int temp,i,j;
    for(i=0;i<size;i++)
    {
        for(j=i;j<size;j++)
        {
            if(arr[j]<arr[i])
            {
                temp=arr[i];
                arr[i]=arr[j];
                arr[j]=temp;
            }
        }
    }
}

void printArray(int arr[],int size)
{
    for(int i=0;i<size;i++)
    {
        printf("%d\n",arr[i]);
    }
}

int main()
{
    int n,i,j;
    printf("Enter size of both arrays: ");
    scanf("%d",&n);
    int a[n],b[n],c[2*n];
```

```
40      scanf("%d",&n);
41      int a[n],b[n],c[2*n];
42      for(i=0;i<n;i++)
43      {
44          a[i]=rand()%100;
45      }
46      for(i=0;i<n;i++)
47      {
48          b[i]=rand()%100;
49      }
50      printf("\nElements of first array-\n");
51      printArray(a,n);
52
53      printf("\nElements of second array-\n");
54      printArray(b,n);
55
56      sort(a,n);
57      sort(b,n);
58
59      printf("\nSorted first array-\n");
60      printArray(a,n);
61
62      printf("\nSorted second array-\n");
63      printArray(b,n);
64
65      clock_t begin = clock();
66
67      for(j=0;j<n;j++)
68      {
69          c[j]=a[j];
70      }
71      for(int k=0;k<n;k++)
72      {
73          c[j+k]=b[k];
74      }
75      printf("\nMerged array is-\n");
76      printArray(c,2*n);
77
78      median(c,2*n);
79
80      clock_t end = clock();
```

Logs & others

Code::Blocks | Search results | Cccc | Build log | Build messages

Start here | merge sort.c | MEDIAN OF TWO SORTED ARRAY AFTER MERGING.c

```
47      {
48          b[i]=rand()%100;
49      }
50      printf("\nElements of first array-\n");
51      printArray(a,n);
52
53      printf("\nElements of second array-\n");
54      printArray(b,n);
55
56      sort(a,n);
57      sort(b,n);
58
59      printf("\nSorted first array-\n");
60      printArray(a,n);
61
62      printf("\nSorted second array-\n");
63      printArray(b,n);
64
65      clock_t begin = clock();
66
67      for(j=0;j<n;j++)
68      {
69          c[j]=a[j];
70      }
71      for(int k=0;k<n;k++)
72      {
73          c[j+k]=b[k];
74      }
75      printf("\nMerged array is-\n");
76      printArray(c,2*n);
77
78      median(c,2*n);
79
80      clock_t end = clock();
81      double time_spent = (double)(end - begin) / CLOCKS_PER_SEC;
82      printf("\nExecution Time : %f seconds\n", time_spent);
83
84
85      return 0;
86  }
87
```

C:\web developement(html.css.js)\MEDIAN OF TWO SORTED ARRAY AFTER MERGING.c