

C:\Users\Admin\OneDrive\Desktop\4th sem\ADA lab\ADA lab programs\Quick Sort\Quick\_sort\_algrtm.cpp - Dev-C++ 5.11



(globals)

Quick\_sort\_algrtm.cpp merge\_sort\_prog.cpp

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<conio.h>
4 #include<time.h>
5 void quicksort(int number[25],int first,int last){
6     int i, j, pivot, temp;
7     if(first<last){
8         pivot=first;
9         i=first;
10        j=last;
11        while(i<j){
12            while(number[i]<=number[pivot]&&i<last)
13                i++;
14            while(number[j]>number[pivot])
15                j--;
16            if(i<j){
17                temp=number[i];
18                number[i]=number[j];
19                number[j]=temp;
20            }
21            temp=number[pivot];
22            number[pivot]=number[j];
23            number[j]=temp;
24            quicksort(number, first, j-1);
25            quicksort(number, j+1, last);
26        }
27    }
28    int main()
29    {
30        clock_t start,end;double tu,t[5];
31        int i,j,temp,count, number[25000];
32        int ch;
33        while(1)
34        {
35            printf("\n1:For manual entry of N value and array elements");
36            printf("\n2:To display time taken for sorting number of elements N in the range 500 to 14500");
37            printf("\n3:To exit");
38            printf("\nEnter your choice:");
39            scanf("%d", &ch);
40            switch(ch)
41            {
42            case 1:printf("Enter number of elements : ");
43                    scanf("%d",&count);
44                    printf("Enter %d elements: ", count);
45                    start=clock();
46                    for(i=0;i<count;i++)
47                        scanf("%d",&number[i]);
48                    quicksort(number,0,count-1);
49                    end=clock();
50                    tu=((double)(end-start))/CLOCKS_PER_SEC;
51                    printf("Order of Sorted elements: ");
```

Compiler Resources Compile Log Debug Find Results

Line: 74 Col: 2 Sel: 0 Lines: 74 Length: 1840 Insert Done parsing in 0 seconds

C:\Users\Admin\OneDrive\Desktop\4th sem\ADA lab\ADA lab programs\Quick Sort\Quick\_sort\_algrtm.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

Quick\_sort\_algrtm.cpp merge\_sort\_prog.cpp

```
28 int i,j,temp,count, number[25000];
29 int ch;
30 while(1)
31 {
32 printf("\n1:For manual entry of N value and array elements");
33 printf("\n2:To display time taken for sorting number of elements N in the range 500 to 14500");
34 printf("\n3:To exit");
35 printf("\nEnter your choice:");
36 scanf("%d", &ch);
37 switch(ch)
38 {
39 case 1:printf("Enter number of elements : ");
40 scanf("%d",&count);
41 printf("Enter %d elements: ", count);
42 start=clock();
43 for(i=0;i<count;i++)
44 | scanf("%d",&number[i]);
45 quicksort(number,0,count-1);
46 end=clock();
47 tu=((double)(end-start))/CLOCKS_PER_SEC;
48 printf("Order of Sorted elements: ");
49 for(i=0;i<count;i++)
50 | printf(" %d",number[i]);
51 printf("\nTime used %f s",tu);
52 break;
53 case 2:
54 count=500;
55 while(count<=14500) {
56 for(i=0;i<count;i++)
57 {
58 number[i]=count-i;
59 }
60 start=clock();
61 quicksort(number,0,count-1);
62 //Dummy loop to create delay
63 for(j=0;j<145000;j++){ temp=38/600;}
64 end=clock();
65 printf("\n Time taken to sort %d numbers is %f Secs",count,(((double)(end-start))/CLOCKS_PER_SEC));
66 count=count+500;
67 }
68 break;
69 case 3: exit(0);
70 }
71 getchar();
72 }
73 return 0;
74 }
```

Compiler Resources Compile Log Debug Find Results

Line: 74 Col: 2 Sel: 0 Lines: 74 Length: 1840 Insert Done parsing in 0 seconds

C:\Users\Admin\OneDrive\Desktop\4th sem\ADA lab\ADA lab programs\Quick Sort\Quick\_sort\_algrtm.exe

```
1:For manual entry of N value and array elements
2:To display time taken for sorting number of elements N in the range 500 to 14500
3:To exit
Enter your choice:1
Enter number of elements : 5
Enter 5 elements: 132 49 5 0 61
Order of Sorted elements: 0 5 49 61 132
Time used 12.393000s
1:For manual entry of N value and array elements
2:To display time taken for sorting number of elements N in the range 500 to 14500
3:To exit
Enter your choice:2
```

```
Time taken to sort 500 numbers is 0.000000 Secs
Time taken to sort 1000 numbers is 0.002000 Secs
Time taken to sort 1500 numbers is 0.004000 Secs
Time taken to sort 2000 numbers is 0.005000 Secs
Time taken to sort 2500 numbers is 0.006000 Secs
Time taken to sort 3000 numbers is 0.009000 Secs
Time taken to sort 3500 numbers is 0.013000 Secs
Time taken to sort 4000 numbers is 0.015000 Secs
Time taken to sort 4500 numbers is 0.020000 Secs
Time taken to sort 5000 numbers is 0.024000 Secs
Time taken to sort 5500 numbers is 0.029000 Secs
Time taken to sort 6000 numbers is 0.034000 Secs
Time taken to sort 6500 numbers is 0.041000 Secs
Time taken to sort 7000 numbers is 0.047000 Secs
Time taken to sort 7500 numbers is 0.056000 Secs
Time taken to sort 8000 numbers is 0.061000 Secs
Time taken to sort 8500 numbers is 0.070000 Secs
Time taken to sort 9000 numbers is 0.077000 Secs
Time taken to sort 9500 numbers is 0.088000 Secs
Time taken to sort 10000 numbers is 0.098000 Secs
Time taken to sort 10500 numbers is 0.105000 Secs
Time taken to sort 11000 numbers is 0.119000 Secs
Time taken to sort 11500 numbers is 0.131000 Secs
Time taken to sort 12000 numbers is 0.138000 Secs
Time taken to sort 12500 numbers is 0.148000 Secs
Time taken to sort 13000 numbers is 0.163000 Secs
Time taken to sort 13500 numbers is 0.175000 Secs
Time taken to sort 14000 numbers is 0.191000 Secs
Time taken to sort 14500 numbers is 0.199000 Secs
```

```
1:For manual entry of N value and array elements
2:To display time taken for sorting number of elements N in the range 500 to 14500
3:To exit
Enter your choice:3
```

```
-----
Process exited after 30.8 seconds with return value 0
Press any key to continue . . .
```

Type here to search



29°C Light rain 11:51 03-06-2021

