

C heap_sort.c X

C heap_sort.c

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<conio.h>
4  #include<time.h>
5
6  clock_t start,end;
7  double cpu_time;
8
9  void heapadj(int a[], int n)
10 {
11     int i,j,item;
12     j=0;
13     item=a[j];
14     i=2*j+1;
15     while(i<=n-1)
16     {
17         if(i+1<=n-1)
18         {
19             if(a[i]<a[i+1])
20             {
21                 i++;
22             }
23         }
24         if(item<a[i])
25         {
26             a[j]=a[i];
27             j=i;
28             i=2*j+1;
29         }
30         else
31         {
32             break;
33         }
34     }
35     a[j]=item;
36 }
37
38 void heapsort(int a[], int n)
```

C heap_sort.c X

C heap_sort.c

```
36 }
37
38 void heapcons(int a[], int n)
39 {
40     int i,j,k,item;
41     for(k=1;k<n;k++)
42     {
43         item=a[k];
44         i=k;
45         j=(i-1)/2;
46         while(i>0 && item>a[j])
47         {
48             a[i]=a[j];
49             i=j;
50             j=(i-1)/2;
51         }
52         a[i]=item;
53     }
54 }
55
56 int heapsort(int a[], int n)
57 {
58     int i,temp;
59     heapcons(a,n);
60     for(i=n-1;i>0;i--)
61     {
62         temp=a[0];
63         a[0]=a[i];
64         a[i]=temp;
65         heapadj(a,i);
66     }
67 }
68
69 int main()
70 {
71     int n,i,a[10000];
72     srand(time(0));
73     printf("enter number of elements:\n");
```

C heap_sort.c X



C heap_sort.c

```
71  int n,i,a[10000];
72  srand(time(0));
73  printf("enter number of elements:\n");
74  scanf("%d", &n);
75  printf("Array elements:\n");
76  for(i=0;i<n;i++)
77  {
78      a[i]=rand()%100;
79      printf("%d ",a[i]);
80  }
81  start= clock();
82  heapsort(a,n);
83  printf("\nsorted array:\n");
84  for(i=0;i<n;i++)
85  {
86      printf("%d ",a[i]);
87  }
88  end = clock();
89  cpu_time = (double)(end - start) / CLOCKS_PER_SEC;
90  printf("\nExecution time for Heap sort = %f ms\n", cpu_time*1000);
91  getch();
92 }
```



C heap_sort.c X

C heap_sort.c

```
73 int n = 5100001;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

powershell + v ^ X

PS C:\Users\muska\OneDrive\Desktop\C programs> gcc heap_sort.c

PS C:\Users\muska\OneDrive\Desktop\C programs> .\a.exe

enter number of elements:

50

Array elements:

60 5 38 70 15 43 72 36 50 85 48 58 58 36 69 75 54 50 75 61 83 62 90 96 82 60 48 81 78 31 4 95 98 30 88 38 83 47 33 3 98 18 26 19 57 94 35 39 68 7

sorted array:

3 4 5 7 15 18 19 26 30 31 33 35 36 36 38 38 39 43 47 48 48 50 50 54 57 58 58 60 60 61 62 68 69 70 72 75 75 78 81 82 83 83 85 88 90 94 95 96 98 98

Execution time for Heap sort = 3.000000 ms

PS C:\Users\muska\OneDrive\Desktop\C programs> gcc heap_sort.c

PS C:\Users\muska\OneDrive\Desktop\C programs> .\a.exe

enter number of elements:

100

Array elements:

86 56 79 36 95 89 4 18 70 69 65 10 29 34 66 8 22 11 21 50 3 38 5 77 34 26 76 14 86 72 11 49 42 7 57 42 61 93 98 83 64 8 57 90 40 5 3 46 23 59 81 15 64 43 32 49 37 39 40 49 49 93 15 80 34 18 86 0 12 91

4 53 6 78 98 19 69 71 0 44 11 61 46 92 94 99 0 95 17 72 26 98 2 39 14 70 37 55 24 2

sorted array:

0 0 0 2 2 3 3 4 4 5 5 6 7 8 8 10 11 11 11 12 14 14 15 15 17 18 18 19 21 22 23 24 26 26 29 32 34 34 34 36 37 37 38 39 39 40 40 42 42 43 44 46 46 49 49 49 49 50 53 55 56 57 57 59 61 61 64 64 65 66 69 69

70 70 71 72 72 76 77 78 79 80 81 83 86 86 86 89 90 91 92 93 93 94 95 95 98 98 98 99

Execution time for Heap sort = 9.000000 ms

PS C:\Users\muska\OneDrive\Desktop\C programs> █