

SIDDHI SINGH

17BIT0028

ITE - 1004

DATA STRUCTURES AND ALGORITHM

HEAP SORT

```
#include <stdio.h>
```

```
intmain()
```

```
{
```

```
intheap[10], no, i, j, c, root, temp;
```

```
int v=0;
```

```
int k=0;
```

```
intcount[10];
```

```
printf("\n Enter no of elements :");
```

```
scanf("%d", &no);
```

```
printf("\n Enter the nos : ");
```

```
for (i = 0; i< no; i++)
```

```
scanf("%d", &heap[i]);
```

```
for (i = 1; i< no; i++)
```

```
{
```

```
    c = i;
```

```
    do
```

```
    {
```

```
        root = (c - 1) / 2;
```

```
        if (heap[root] > heap[c])    /* to create MAX heap array */
```

```
        {
```

```
            v++;
```

```
            count[k]=v;
```

```
            k++;
```

```
                temp = heap[root];
```

```
                heap[root] = heap[c];
```

```
                heap[c] = temp;
```

```
        }
```

```
        c = root;
```

```
    } while (c != 0);
```

```
}
```

```

printf("Heap array : ");

for (i = 0; i < no; i++)
printf("%d\t", heap[i]);

for (j = no - 1; j >= 0; j--)
{
    temp = heap[0];
    heap[0] = heap[j];    /* swap max element with rightmost leaf element
    */
    heap[j] = temp;
    root = 0;
    do
    {
        c = 2 * root + 1;    /* left node of root element */
        if ((heap[c] > heap[c + 1]) && c < j-1)
            c++;
        if (heap[root] > heap[c] && c < j)    /* again rearrange to
        max heap array */
        {
            temp = heap[root];
            heap[root] = heap[c];
            heap[c] = temp;
        }

        root = c;
    } while (c < j);
}

printf("\n The sorted array is : ");
for (i = 0; i < no; i++)
printf("\t %d", heap[i]);
printf("\n the number of swap in each method is:\n");
i = 0;
while (count[i] != NULL)
{
    printf("%d\n", count[i]);
    i++;
}
return 0;
}

```

Untitled1.c - Code::Blocks 16.01

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoryBlocks Settings Help

Management

Projects Symbols Files

Workspace

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int heap[10], no, i, j, c, root, temp;
6     int v=0;
7     int k=0;
8     int count[10];
9
10    printf("\n Enter no of elements :");
11    scanf("%d", &no);
12    printf("\n Enter the nos : ");
13    for (i = 0; i < no; i++)
14        scanf("%d", &heap[i]);
15    for (i = 1; i < no; i++)
16    {
17        c = i;
18        do
19        {
20            root = (c - 1) / 2;
21            if (heap[root] > heap[c]) /* to create MAX heap array */
22            {
23                v++;
24                count[k]=v;
25                k++;
26                temp = heap[root];
27                heap[root] = heap[c];
28                heap[c] = temp;
29            }
30            c = root;
31        } while (c != 0);
32    }
```

C:\Users\Rites\Documents\Untitled1.c Windows (CR+LF) default Line 72, Column 1 Insert Read/Write default 11:00 PM 20-Mar-18

Untitled1.c - Code::Blocks 16.01

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoryBlocks Settings Help

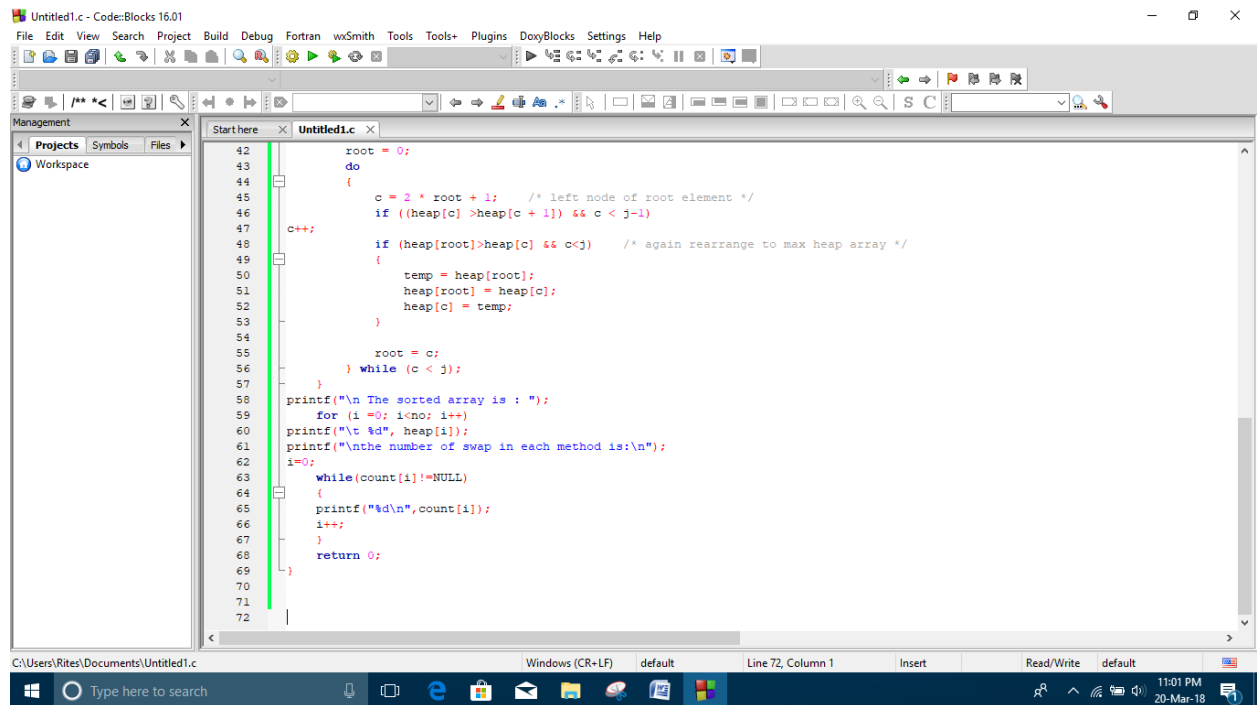
Management

Projects Symbols Files

Workspace

```
26     temp = heap[root];
27     heap[root] = heap[c];
28     heap[c] = temp;
29 }
30 c = root;
31 } while (c != 0);
32 }
33
34 printf("Heap array : ");
35 for (i = 0; i < no; i++)
36     printf("%d\t", heap[i]);
37 for (j = no - 1; j >= 0; j--)
38 {
39     temp = heap[0];
40     heap[0] = heap[j]; /* swap max element with rightmost leaf element */
41     heap[j] = temp;
42     root = 0;
43     do
44     {
45         c = 2 * root + 1; /* left node of root element */
46         if ((heap[c] > heap[c + 1]) && c < j-1)
47             c++;
48         if (heap[root] > heap[c] && c < j) /* again rearrange to max heap array */
49         {
50             temp = heap[root];
51             heap[root] = heap[c];
52             heap[c] = temp;
53         }
54         root = c;
55     } while (c < j);
56 }
```

C:\Users\Rites\Documents\Untitled1.c Windows (CR+LF) default Line 72, Column 1 Insert Read/Write default 11:01 PM 20-Mar-18



```
42 root = 0;
43 do
44 {
45     c = 2 * root + 1; /* left node of root element */
46     if ((heap[c] > heap[c + 1]) && c < j-1)
47         c++;
48     if (heap[root] > heap[c] && c < j) /* again rearrange to max heap array */
49     {
50         temp = heap[root];
51         heap[root] = heap[c];
52         heap[c] = temp;
53     }
54     root = c;
55 } while (c < j);
56
57 printf("\n The sorted array is : ");
58 for (i = 0; i < no; i++)
59     printf("%d ", heap[i]);
60 printf("\n the number of swap in each method is:\n");
61 i = 0;
62 while (count[i] != NULL)
63 {
64     printf("%d\n", count[i]);
65     i++;
66 }
67 return 0;
68
69
70
71
72
```

OUTPUT

Enter no of elements :10

Enter the nos : 1

2
3
4
5
6
7
8
9
0

Heap array : 0 1 3 4 2 6 7 8 9

5

The sorted array is : 9 8 7 6 5 4 3

2 1 0

the number of swap in each method is:

1
2
3

...Program finished with exit code 0

Press ENTER to exit console.