Logs & others

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

Start here | heap sort.c

```c
#include <stdio.h>
  #include <time.h>
  void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
  }

  void heapify(int arr[], int n, int i) {
    int largest = i;
    int left = 2 * i + 1;
    int right = 2 * i + 2;

    if (left < n && arr[left] > arr[largest])
      largest = left;

    if (right < n && arr[right] > arr[largest])
      largest = right;
    if (largest != i) {
      swap(&arr[i], &arr[largest]);
      heapify(arr, n, largest);
    }
  }
  void heapSort(int arr[], int n) {
    for (int i = n / 2 - 1; i >= 0; i--)
      heapify(arr, n, i);
    for (int i = n - 1; i >= 0; i--) {
      swap(&arr[0], &arr[i]);
      heapify(arr, i, 0);
    }
  }
  void printArray(int arr[], int n) {
    for (int i = 0; i < n; ++i)
      printf("%d ", arr[i]);
    printf("\n");
  }
  int main()
  {
      int n, i, j, temp,c;
      printf("enter the value of N:");
      scanf("%d",&n);
```

Logs & others

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

Start here | heap sort.c

```
17    if (right < n && arr[right] > arr[largest])
18      largest = right;
19    if (largest != i) {
20      swap(&arr[i], &arr[largest]);
21      heapify(arr, n, largest);
22    }
23  }
24  void heapSort(int arr[], int n) {
25    for (int i = n / 2 - 1; i >= 0; i--)
26      heapify(arr, n, i);
27    for (int i = n - 1; i >= 0; i--) {
28      swap(&arr[0], &arr[i]);
29      heapify(arr, i, 0);
30    }
31  }
32  void printArray(int arr[], int n) {
33    for (int i = 0; i < n; ++i)
34      printf("%d ", arr[i]);
35    printf("\n");
36  }
37  int main()
38  {
39      int n, i, j, temp,c;
40      printf("enter the value of N:");
41      scanf("%d",&n);
42    int arr[10];
43      printf("Genrating random nos.:\n");
44
45  for (c = 0; c < n; c++){
46    arr[c]=rand()%100 +1;
47    printf("%d\t\n",arr[c]);
48   }
49    clock_t begin = clock();
50   heapSort(arr, n);
51    clock_t end = clock();
52    printf("Sorted array is \t");
53    printArray(arr, n);
54  double time_spent = (double)(end - begin) / CLOCKS_PER_SEC;
55  printf("\n\nEXECUTION TIME : %.10fseconds\n", time_spent);
56  }
57
```

```
"C:\web developement(html,css,js)\heap sort.exe"
enter the value of N:25
Genrating random nos.:
42
68
35
1
70
25
79
59
63
65
6
46
82
28
62
92
96
43
28
0
Sorted array is         0 0 0 0 0 1 6 25 28 35 38 42 43 46 47 59 62 15130 65 68 70 79 82 92 96 15130 1774496 4199367 4225392 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

EXECUTION TIME : 15.0650000000seconds

Process returned -1073741819 (0xC0000005)   execution time : 20.436 s
Press any key to continue.
```