



```
main.c
1 #include <stdio.h>
2 void heapify(int arr[], int n, int i) {
3     int largest = i;
4     int left = 2 * i + 1;
5     int right = 2 * i + 2;
6
7     if (left < n && arr[left] > arr[largest])
8         largest = left;
9
10    if (right < n && arr[right] > arr[largest])
11        largest = right;
12
13    if (largest != i) {
14        int temp = arr[i];
15        arr[i] = arr[largest];
16        arr[largest] = temp;
17        heapify(arr, n, largest);
18    }
19 }
20 void buildHeap(int arr[], int n) {
21     for (int i = n / 2 - 1; i >= 0; i--)
22         heapify(arr, n, i);
23 }
24 void printArray(int arr[], int n) {
25     for (int i = 0; i < n; ++i)
26         printf("%d ", arr[i]);
27     printf("\n");
28 }
29 int main() {
30     int arr[] = {12, 11, 13, 5, 6, 7};
31     int n = sizeof(arr) / sizeof(arr[0]);
32     buildHeap(arr, n);
33     printf("Heap array: \n");
34     printArray(arr, n);
35     return 0;
36 }
37
```

Share Run

Output Clear

```
/tmp/hWQFi881Kc.o
Heap array:
13 11 12 5 6 7

=== Code Execution Successful ===
```

main.c

Run

Share

Clear

Output

```
1 #include <stdio.h>
2 void heapify(int arr[], int n, int i) {
3     int largest = i;
4     int left = 2 * i + 1;
5     int right = 2 * i + 2;
6     if (left < n && arr[left] > arr[largest])
7         largest = left;
8     if (right < n && arr[right] > arr[largest])
9         largest = right;
10    if (largest != i) {
11        int temp = arr[i];
12        arr[i] = arr[largest];
13        arr[largest] = temp;
14        heapify(arr, n, largest);
15    }
16 }
17 void heapSort(int arr[], int n) {
18     for (int i = n / 2 - 1; i >= 0; i--)
19         heapify(arr, n, i);
20     for (int i = n - 1; i > 0; i--) {
21         int temp = arr[0];
22         arr[0] = arr[i];
23         arr[i] = temp;
24         heapify(arr, i, 0);
25     }
26 }
27 void printArray(int arr[], int n) {
28     for (int i = 0; i < n; ++i)
29         printf("%d ", arr[i]);
30     printf("\n");
31 }
32 int main() {
33     int arr[] = {12, 11, 13, 5, 6, 7};
34     int n = sizeof(arr) / sizeof(arr[0]);
35     printf("Original array: \n");
36     printArray(arr, n);
37     heapSort(arr, n);
38     printf("Sorted array: \n");
39     printArray(arr, n);
40     return 0;
41 }
42
```

/tmp/Qi19bud9hS.o

Original array:
12 11 13 5 6 7
Sorted array:
5 6 7 11 12 13

=== Code Execution Successful ===

FREE AI Code Generator: Gener...

Online C Compiler - Programiz

programiz.com/c-programming/online-compiler/

Paused

Programiz

C Online Compiler

LOOKING TO LEARN PROGRAMMING?

Start your programming journey with Programiz AT NO COST.

Programiz PRO >

main.c

Run

Clear

```
1 #include <stdio.h>
2 void heapify(int arr[], int n, int i) {
3     int smallest = i;
4     int left = 2 * i + 1;
5     int right = 2 * i + 2;
6     if (left < n && arr[left] < arr[smallest])
7         smallest = left;
8     if (right < n && arr[right] < arr[smallest])
9         smallest = right;
10    if (smallest != i) {
11        int temp = arr[i];
12        arr[i] = arr[smallest];
13        arr[smallest] = temp;
14        heapify(arr, n, smallest);
15    }
16 }
17 void buildMinHeap(int arr[], int n) {
18     for (int i = n / 2 - 1; i >= 0; i--)
19         heapify(arr, n, i);
20 }
21 void heapSort(int arr[], int n) {
22     buildMinHeap(arr, n);
23
24     for (int i = n - 1; i > 0; i--) {
25         int temp = arr[0];
26         arr[0] = arr[i];
27         arr[i] = temp;
28         heapify(arr, i, 0);
29     }
30 }
31 void printArray(int arr[], int n) {
32     for (int i = 0; i < n; ++i)
33         printf("%d ", arr[i]);
34     printf("\n");
35 }
36 int main() {
37     int arr[] = {12, 11, 13, 5, 6, 7};
38     int n = sizeof(arr) / sizeof(arr[0]);
39     printf("Original array: \n");
40     printArray(arr, n);
41     heapSort(arr, n);
42     printf("Sorted array: \n");
43     printArray(arr, n);
```

Output

```
/tmp/n5kTvv1Hve.o
Original array:
12 11 13 5 6 7
Sorted array:
13 12 11 7 6 5

=== Code Execution Successful ===
```

Athletics

Final result

Search

10:01 AM

8/5/2024