Q.

Ans.

**Heap Sort :-** Heap sort is a comparison-based sorting technique based on Binary Heap data structure. It is similar to selection sort where we first find the minimum element and place the minimum element at the beginning. We repeat the same process for the remaining elements.

A [Binary Heap](https://www.geeksforgeeks.org/binary-heap/) is a Complete Binary Tree where items are stored in a special order such that the value in a parent node is greater(or smaller) than the values in its two children nodes. The former is called max heap and the latter is called min-heap.

**Applications :-**

1. Sort a nearly sorted (or K sorted) array.
2. K largest (or smallest) elements in an array.

**Heap Sort Algorithm :-**

HEAP\_SORT(ARR, N)

* **Step 1 :** [Build Heap H]

Repeat for i = 0 to N – 1

CALL INSERT\_HEAP(ARR, N, ARR[i])

[END OF LOOP]

* **Step 2 :** Repeatedly Delete the root element

Repeat while N > 0

CALL Delete\_Heap(ARR, N, VAL)

SET N = N + 1

[END OF LOOP]

* **Step 3 :** END

**Time Complexity for Heap Sort :-**

Worst Case Time Complexity: **O(n\*log n)**

Best Case Time Complexity: **O(n\*log n)**

Average Time Complexity: **O(n\*log n)**

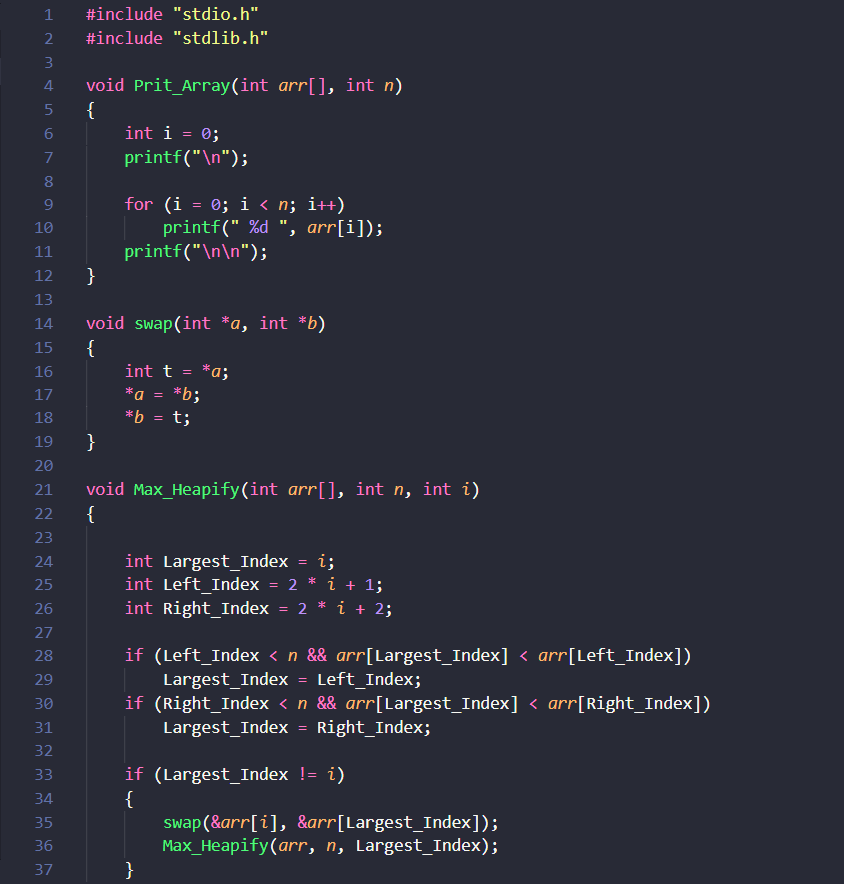
**Space Complexity for Heap Sort :-**

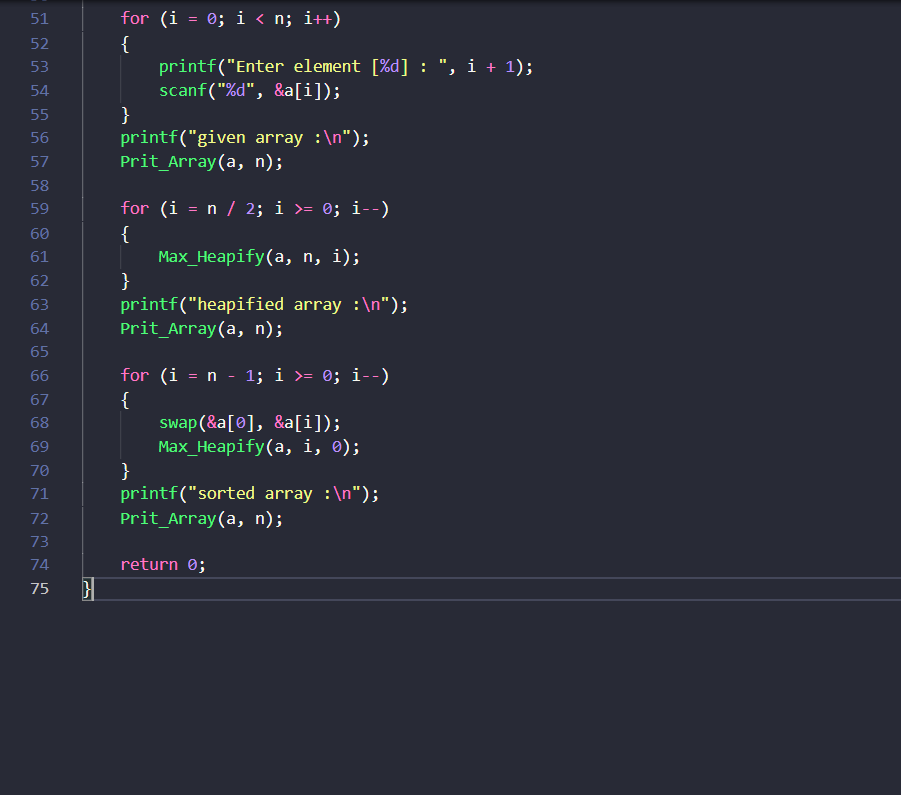
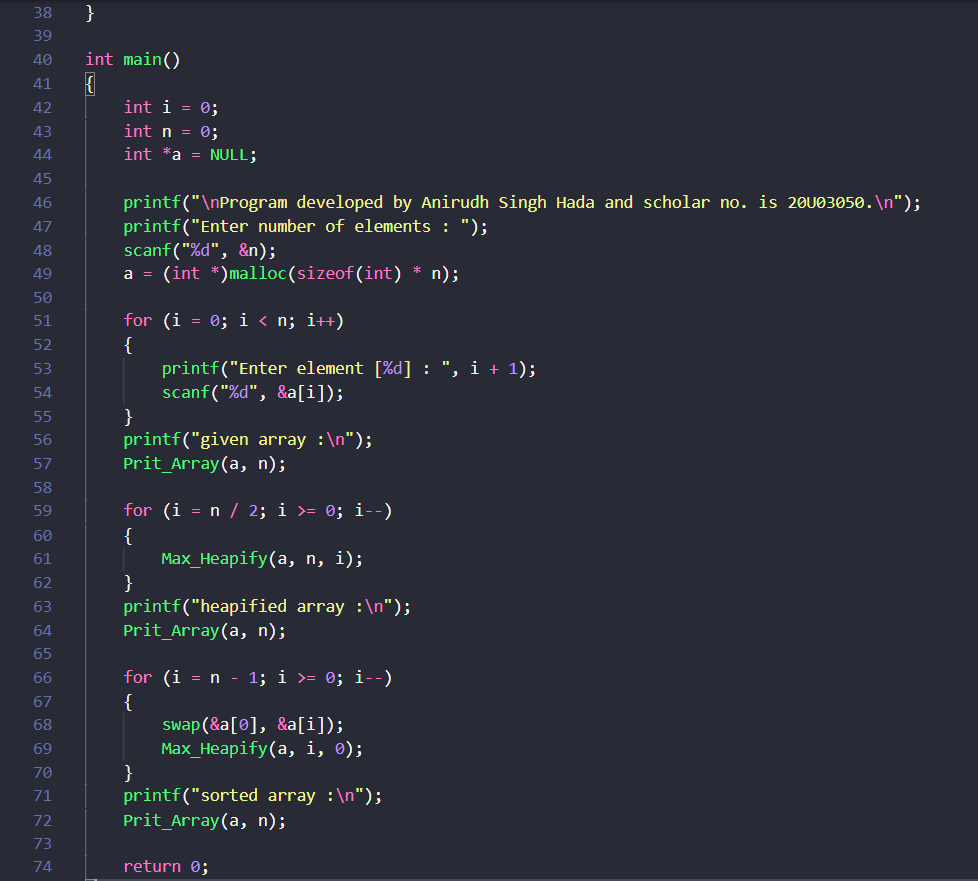
The space complexity for heap sort is **O(1)**.

* Heap sort is not a stable sort, and requires a constant space for sorting a list.
* Heap sort is very fast and is widely used for sorting.

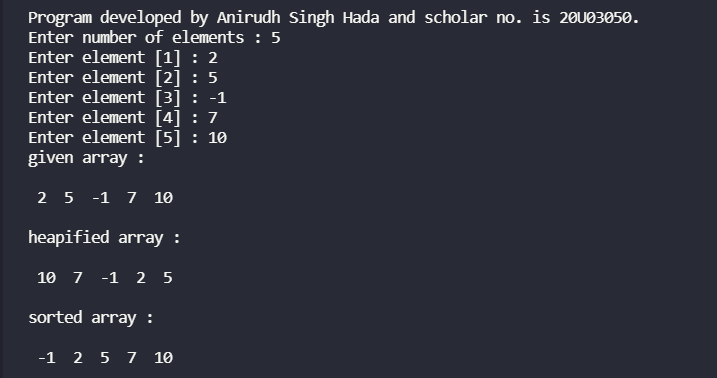
**Implementation of Heap Sort :-**

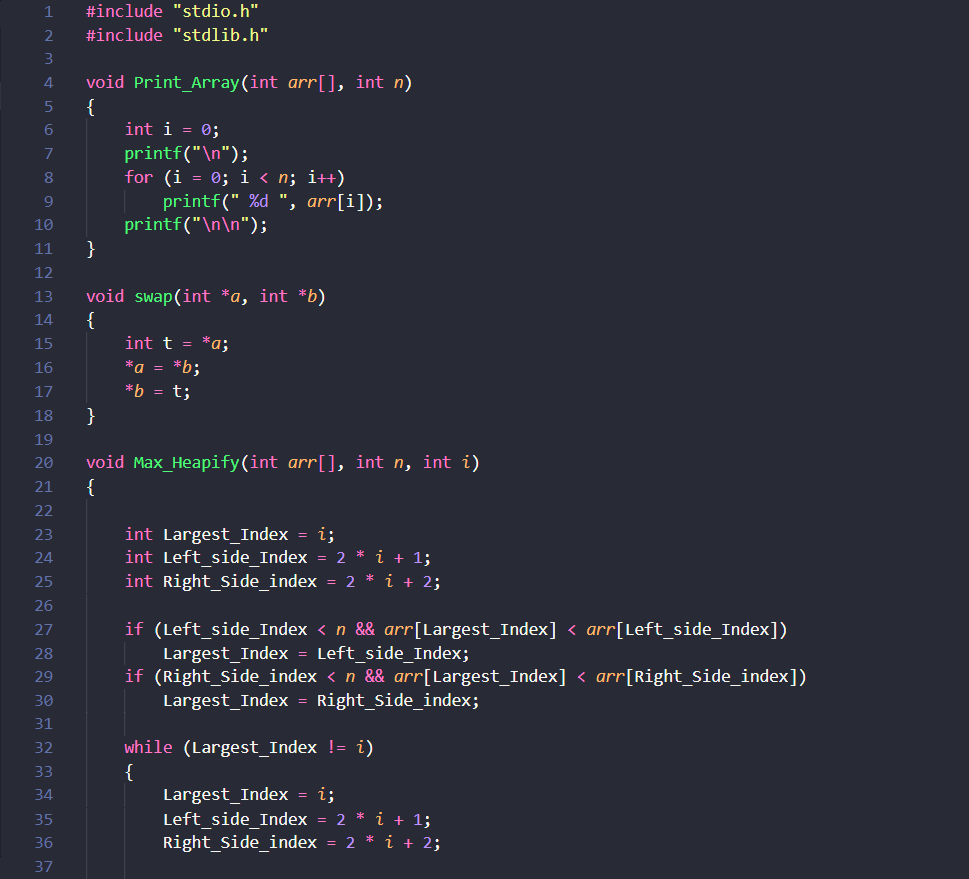
1. **Recursive Code :-**

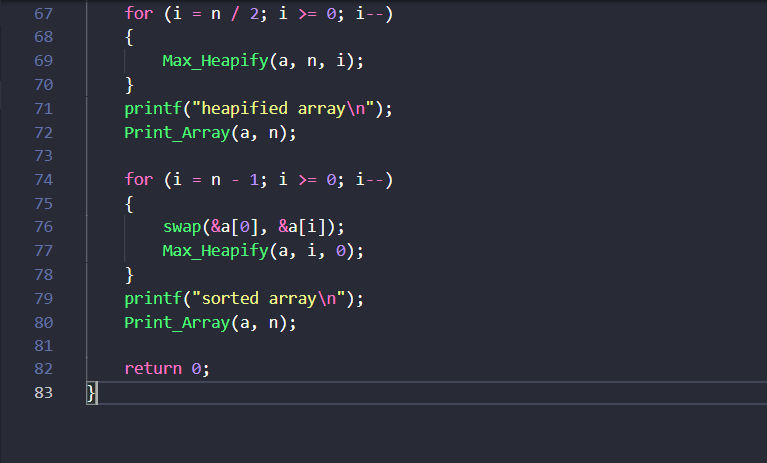
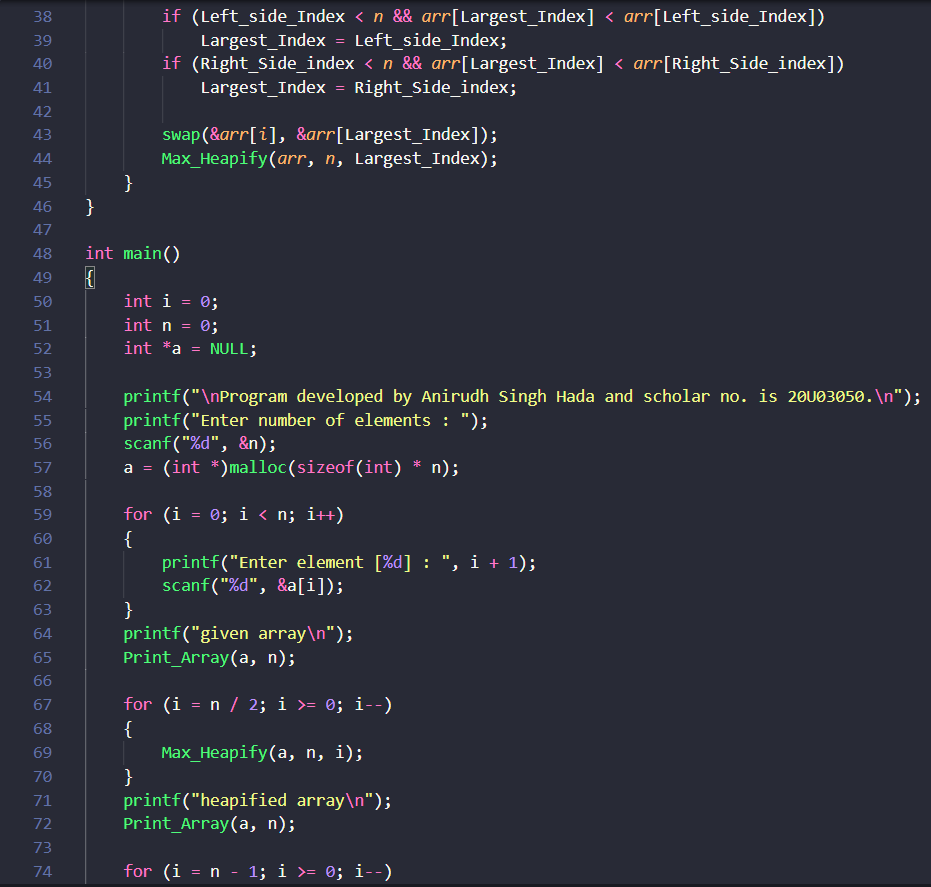
****

****

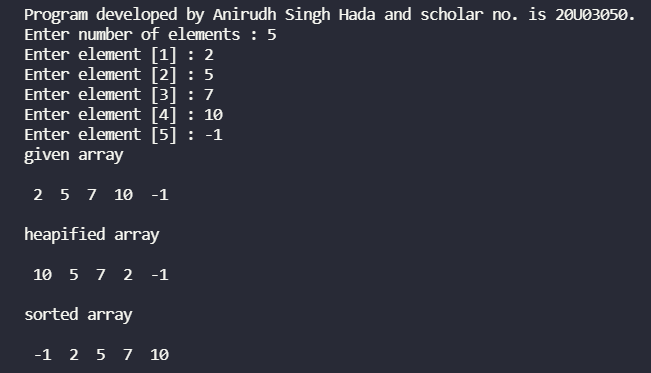
**Output :-**

****

1. **Iterative Code :-** 



**Output :-**

****