

Program

```
#include <stdio.h>
#define SIZE 50
void heapify(int arr[], int n, int i) {
    int temp, 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) {
        temp = arr[i];
        arr[i] = arr[largest];
        arr[largest] = temp;
        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--) {
        int temp = arr[0];
        arr[0] = arr[i];
        arr[i] = temp;
        heapify(arr, i, 0);
    }
}

int main() {
    int arr[SIZE], n, i;

    printf("Enter size of the array: ");
    scanf("%d", &n);
    printf("Enter %d elements:\n", n);
    for (i = 0; i < n; i++)
        scanf("%d", &arr[i]);

    heapSort(arr, n);

    printf("\nSorted array: ");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }

    return 0;
}
```

HEAP SORT

Aim:

To implement Heap Sort algorithm.

Algorithm:

1. Start

2. Define SIZE as 50

3. Define function void heapify(int arr[] , int n , int i)

```
Define int temp
Initialize largest = i, left = 2 * i + 1, right = 2 * i + 2
Check if ( left < n and arr[left] > arr[largest])
    Set largest = left
Check if ( right < n and arr[right] > arr[largest])
    Set largest = right
Check if ( largest != i)
    Set temp = arr[i]
    arr[i] = arr[largest]
    arr[largest] = temp
    Call heapify(arr, n, largest)
```

4. Define function void heapsort(int arr[], int n)

```
Begin for loop from i = n/2 - 1 to i >= 0
    Call heapify(arr , n, i)
    Decrement i by 1
Begin for loop from i = n - 1 to i > 0
    Set temp = arr[0]
    arr[0] = arr[i]
    arr[i] = temp
    Call heapify(arr , i, 0)
```

5. In main()

```
Declare int arr[SIZE], n , i
Create an array 'arr' of size n
Read n elements into the array
Call heapsort(arr, n)
Begin for loop from i = 0 to i < n
    Display arr[i]
```

Output

Enter size of the array: 10

Enter 10 elements:

67

89

3

12

65

38

22

5

17

95

Sorted array: 3 5 12 17 22 38 65 67 89 95

6. Stop

Result:

Program has been executed successfully and obtained the output.