

Aim:

Write a program to sort (ascending order) the given elements using heap sort technique.

Note: Do use the printf() function with a newline character (\n).

Source Code:**HeapSortMain.c**

```
#include<stdio.h>
int disy(int [],int );
int heapsort(int [],int );
int heapify(int [],int ,int );
void main()
{
    int arr[15],i,n;
    printf("Enter array size : ");
    scanf("%d", &n);
    printf("Enter %d elements : ", n);
    for (i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    printf("Before sorting the elements are : ");
    disy(arr,n);
    heapsort(arr,n);
    printf("After sorting the elements are : ");
    disy(arr,n);
}
int disy(int arr[15],int n)
{
    int i;
    for(i=0;i<n;i++)
    {
        printf("%d ",arr[i]);
    }
    printf("\n");
}
int heapsort(int arr[15],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 heapify(int arr[15],int n,int i)
```

```

{
    int largest=i;
    int l=2*i+1;
    int r=2*i+2;
    if(l<n && arr[l]>arr[largest])
        largest=l;
    if(r<n && arr[r]>arr[largest])
        largest=r;
    if(largest!=i)
    {
        int temp=arr[i];
        arr[i]=arr[largest];
        arr[largest]=temp;
        heapify(arr,n,largest);
    }
}

```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter array size : 5
Enter 5 elements : 23 54 22 44 12
Before sorting the elements are : 23 54 22 44 12
After sorting the elements are : 12 22 23 44 54

Test Case - 2
User Output
Enter array size : 6
Enter 6 elements : 12 65 23 98 35 98
Before sorting the elements are : 12 65 23 98 35 98
After sorting the elements are : 12 23 35 65 98 98

Test Case - 3
User Output
Enter array size : 4
Enter 4 elements : -23 -45 -12 -36
Before sorting the elements are : -23 -45 -12 -36
After sorting the elements are : -45 -36 -23 -12

Test Case - 4
User Output
Enter array size : 6
Enter 6 elements : 1 -3 8 -4 -2 5
Before sorting the elements are : 1 -3 8 -4 -2 5
After sorting the elements are : -4 -3 -2 1 5 8