MAX-HEAP SORT

Exp. No.:

AIM:

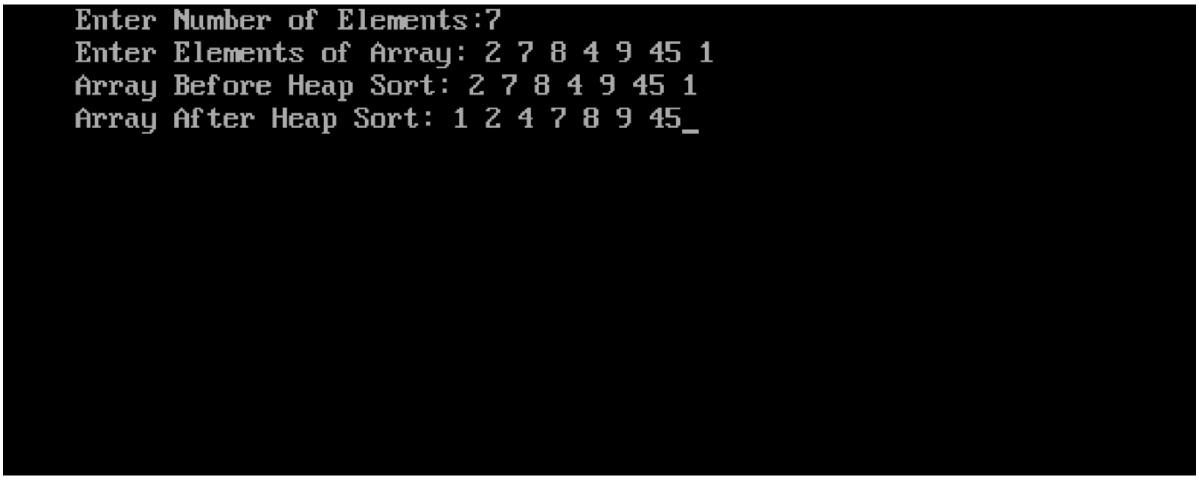
ALGORITHM:

PROGRAM:

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>

void heapify(int *arr, int n, int i)
{
   int largest, temp, left, right;
   largest = i;
   left = 2 * i + 1;
   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)
{
   int i, temp;
   for (i = n / 2 - 1; i >= 0; i--)
   {
      heapify(arr, n, i);
   }
   for (i = n - 1; i >= 0; i--)
```

```
    {
        temp = arr[0];
        arr[0] = arr[i];
        arr[i] = temp;

        heapify(arr, i, 0);
    }
}

int main()
{
    int *arr, n, i;
    printf("Enter Number of Elements:");
    scanf("%d", &n);
    arr = (int *)malloc(n * sizeof(int));
    printf("Enter Elements of Array: ");
    for (i = 0; i < n; i++)
    {
        scanf("%d", &arr[i]);
    }
    printf("Array Before Heap Sort:");
    for (i = 0; i < n; i++)
    {
        printf(" %d", arr[i]);
    }
    heapSort(arr, n);
    printf("\nArray After Heap Sort:");
    for (i = 0; i < n; i++)
    {
        printf(" %d", arr[i]);
    }
    getch();
    clrscr();
    return 0;
}
```

OUTPUT:



```
Enter Number of Elements:7
Enter Elements of Array: 2 7 8 4 9 45 1
Array Before Heap Sort: 2 7 8 4 9 45 1
Array After Heap Sort: 1 2 4 7 8 9 45_
```

RESULT: