**WAP to create Heap from the given Array**

#include <stdio.h>

void swap(int \*a, int \*b)

{

int temp = \*b;

\*b = \*a;

\*a = temp;

}

void printArray(int arr[], int n)

{

printf("\n\nRepresenation of array in max-heap: ");

int i;

for (i = 0; i < n; ++i)

{

printf("%d \t", arr[i]);

}

printf("\n");

}

void heapify(int arr[], int n, int i)

{

int largest = i; // Initialize largest as root

int leftChild = 2 \* i + 1; // left child = 2\*i + 1

int rightChild = 2 \* i + 2; // right child = 2\*i + 2

// If left child is greater than root

if (leftChild < n && arr[leftChild] > arr[largest])

largest = leftChild;

// If right child is greater than new largest

if (rightChild < n && arr[rightChild] > arr[largest])

largest = rightChild;

// If largest is not the root

if (largest != i)

{

swap(&arr[i], &arr[largest]);

heapify(arr, n, largest);

}

}

// Function to build a Max-Heap from a given array

void buildHeap(int arr[], int n)

{

// Index of last non-leaf node

int i;

int lastNonLeafNode = (n / 2) - 1;

for (i = lastNonLeafNode; i >= 0; i--)

{

heapify(arr, n, i);

}

}

void main()

{

int arr[10],n,i;

printf("\n Array Size:");

scanf("%d",&n);

printf("\n Enter Array elements: ");

for (i=0;i<n;i++)

scanf("%d",&arr[i]);

buildHeap(arr, n);

printArray(arr, n);

}