**AIM : Implement the Heap/Shell sort algorithm implemented in Java demonstrating heap/shell data structure with modularity of programming language.**

**PROGRAM :-**

import java.util.\*;

public class Base{

public void sort(int arr[]){

int n = arr.length;

//Rearrange array (building heap)

for(int i=n / 2 - 1;i>=0;i--){

heapify(arr,n,i);

}

//Extract elements from heap one by one

for(int i=n-1;i>=0;i--){

//Current root moved to the end

int temp = arr[0];

arr[0] = arr[i];

arr[i] = temp;

heapify(arr,i,0); // calling max heapify on the heap reduced

}

}

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

int max = i; //Initiliaze max as root

int leftChild = 2 \* i + 1;

int rightChild = 2 \* i + 2;

//If left child is greater than root

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

max = leftChild;

//If right child is greater than root

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

max = rightChild;

if(max != i){

int swap = arr[i];

arr[i] = arr[max];

arr[max] = swap;

//heapify the affected sub-tree recursively

heapify(arr,n,max);

}

}

//print size of array n using utility function

static void display(int arr[]){

int n = arr.length;

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

System.out.print(arr[i]+" ");

}

System.out.println();

}

//Driver Code

public static void main(String args[]){

Scanner sc = new Scanner(System.in);

int n;

System.out.println("Enter the number of elements to be sorted:");

n = sc.nextInt();

int arr[] = new int [n];

System.out.println("Enter "+n+" integer elements");

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

arr[i] = sc.nextInt();

Base bs = new Base();

System.out.println("Original array: ");

display(arr);

bs.sort(arr);

System.out.println("Array after Heap Sort :");

display(arr);

}

}