M**A**XH**E**AP

#include<bits/stdc++.h>

using namespace std;

void heapify(vector <int> &ht,int i)

{

int size=ht.size();

int largest=i;

int l=2\*i+1;

int r=2\*i+2;

if(l<size && ht[l]>ht[largest])

largest=l;

if(r<size && ht[r]>ht[largest])

largest=r;

if(largest!=i)

{

swap(ht[i],ht[largest]);

heapify(ht,largest);

}

}

void printarray(vector <int> &ht)

{

for(int i=0;i<ht.size();i++)

cout<<ht[i]<<" ";

}

int main()

{

int n,x;

vector<int> heaptree;

cin>>n;

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

{

cin>>x;

heaptree.push\_back(x);

}

int size=heaptree.size();

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

{

heapify(heaptree,i);

}

printarray(heaptree);

return 0;

}

M**I**NH**E**AP

#include<bits/stdc++.h>

using namespace std;

void heapify(vector <int> &ht,int i)

{

int size=ht.size();

int minimum=i;

int l=2\*i+1;

int r=2\*i+2;

if(l<size && ht[l]<ht[minimum])

minimum=l;

if(r<size && ht[r]<ht[minimum])

minimum=r;

if(minimum!=i)

{

swap(&ht[i],&ht[minimum]);

heapify(ht,minimum);

}

}

void printarray(vector <int> &ht)

{

for(int i=0;i<ht.size();i++)

cout<<ht[i]<<" ";

}

int main()

{

int n,x;

vector<int> heaptree;

cin>>n;

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

{

cin>>x;

heaptree.push\_back(x);

}

int size=heaptree.size();

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

{

heapify(heaptree,i);

}

printarray(heaptree);

return 0;

}