#include<iostream>

#include<math.h>

using namespace std;

#define maxn 1000

int x[maxn];

int parent(int i){

return int(i/2);

}

int left(int i){

return 2\*i;

}

int right(int i){

return 2\*i+1;

}

void max\_heap(int x[],int i,int size){

bool s=true;

int largest;

for (int k=1;k<size/2;k++){

if(x[k]< x[2\*k] || x[k]<x[2\*k+1]){

s=false;

}

}

if (s==true) return;

else{

if(i>=size) return ;

int l=left(i);

int r=right(i);

if (l<=size && x[l]>x[i]){

largest=l;

}

else

{

largest=i;

}

if (r<=size && x[r]>x[largest]){

largest=r;

}

if (largest!=i) { int s=x[i];x[i]=x[largest];x[largest]=s;}

}

max\_heap(x,largest,size);

}

void build(int x[],int size){

int heapsize=size;

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

max\_heap(x,i,size);

}

int main(){

x[1]=4;

x[2]=1;

x[3]=3;

x[4]=2;

x[5]=16;

x[6]=9;

x[7]=10;

x[8]=14;

x[9]=8;

x[10]=7;

build(x,10);

for (int i=1;i<=10;i++)

cout<<x[i]<<" ";

return 0;

}