#include<bits/stdc++.h>

using namespace std;

class median{

public:

vector<int> v;

void add(int num){

v.push\_back(num);

int ci = v.size()-1;

upheapify(ci);

}

void upheapify(int ci){

if(ci==0){

return;

}

int pi = (ci-1)/2;

if(v[ci]<v[pi]){

swap(v[ci],v[pi]);

upheapify(pi);

}

}

void remove\_top(){

if(v.size()==0){

cout<<"Underflow"<<endl;

return;

}

int ci = v.size()-1;

int pi = 0;

int val = v[0];

swap(v[ci],v[pi]);

v.pop\_back();

downheapify(0);

cout<<val<<endl;

}

void downheapify(int pi){

// if(pi==v.size()-1){

// return;

// }

int li = (2\*pi) + 1;

int ri = (2\*pi) + 2;

int mini = pi;

if(li<v.size() && v[li]<v[pi]){

mini = li;

}

if(li<v.size() && v[ri]<v[mini]){

mini = ri;

}

if(mini!=pi){

swap(v[mini],v[pi]);

downheapify(mini);

}

}

int peek(){

if(v.size()==0){

cout<<"Underflow"<<endl;

return -1;

}

return v[0];

}

int size\_heap(){

return v.size();

}

};

int main(){

string s;

cin>>s;

median med;

while(s!="quit"){

int num;

if(s=="remove"){

med.remove\_top();

}

else if(s=="peek"){

if(med.peek()!=-1)

cout<<med.peek()<<endl;

}

else if(s=="size"){

cout<<med.size\_heap()<<endl;

}

else if(s=="add"){

cin>>num;

med.add(num);

}

else{

//do nothing

}

cin>>s;

}

}