C2-19-197188

#include<iostream>

**using** **namespace** std;

**typedef** **struct** btnode \* btptr;

#define null NULL

**struct** btnode{

btptr lc=null,rc=null;

**char** data;

**int** at=**0**,dt=**0**;

};

**void** **insert**(btptr &t,**char** k)

{

**if**(t==null){

t=**new**(btnode);

t->data=k;

t->rc=t->lc=null;

}

**char** x;

cin>>x;

**if**(x!='.')insert(t->lc,x);

cin>>x;

**if**(x!='.')insert(t->rc,x);

}

**struct** queue{

**int** size=**100**,f=-**1**,r=-**1**;

btptr elements[**100**];

**void** **enq**(btptr t)

{

**if**((r+**1**)%size==f)**return**;

**else**{

**if**(f==-**1**)f=r=**0**;

**else** r=(r+**1**)%size;

elements[r]=t;

}

}

btptr **deq**()

{

**if**(f==-**1**)**return** null;

**else**{

btptr t;

t=elements[f];

**if**(f==r)f=r=-**1**;

**else** f=(f+**1**)%size;

**return** t;

}

}

};

**int** **length**(**struct** queue q)

{

**int** x=**0**;

**while**(q.f>=**0**){

x++;

q.deq();

}

**return** x;

}

**void** **printq**(**struct** queue q)

{

**while**(q.f>=**0**)cout<<q.deq()->data<<" ";

cout<<endl;

}

**void** **busy**(btptr t,**struct** queue q,**int** n)

{

**if**(t==null)**return**;

**if**(t->lc==null && t->rc ==null){

q.enq(t);

**if**(length(q)==n)printq(q);

**return**;

}

q.enq(t);

busy(t->lc,q,n);

busy(t->rc,q,n);

}

**void** **time**(btptr &t)

{

**static** **int** curtime=**0**;

**if**(t==null)**return**;

t->at=curtime++;

time(t->lc);

time(t->rc);

t->dt=curtime++;

}

**int** **max**(**int** a,**int** b)

{

**return** (a>b)?a:b;

}

**int** **platforms**(btptr t)

{

**if**(!t)**return** **0**;

**if**(!t->lc && !t->rc)**return** **1**;

**return** **1** + max(platforms(t->lc),platforms(t->rc));

}

**void** **stay**(btptr t,**int** x)

{

**if**(!t)**return**;

**if**(t->dt-t->at==x)cout<<t->data<<" ";

stay(t->lc,x);

stay(t->rc,x);

}

**int** **main**()

{

btptr t=null;

**char** x;

cin>>x;

insert(t,x);

**int** hrs;cin>>hrs;

time(t);

cout<<platforms(t)<<endl;

stay(t,hrs);cout<<endl;

**struct** queue q;

busy(t,q,platforms(t));

**return** **0**;

}

// ABCDE...F..G..HIJ...K.L.. 3

INPUT:

ABCDE...F..G..HIJ...K.L.. 3

OUTPUT:

