**State.java**

**public** **class** State {

**int** blocks;

**int** on[][];

**int** ontable[];

**int** clear[];

**int** hold[];

**int** arm;

State(**int** n,String desc)

{

blocks =n;

on = **new** **int**[n][n];

ontable = **new** **int**[n];

clear = **new** **int**[n];

hold = **new** **int**[n];

arm =-1;

setState(desc);

}

**void** setState(String desc)

{

**int** i;

String subG[] = desc.split("['^']+");

**for**(i=0;i<subG.length;i++)

{

String ele[] = subG[i].split("[() ]+");

**if**(ele[1].equals("on"))

on[(**int**)ele[2].charAt(0)%97][(**int**)ele[3].charAt(0)%97] = 1;

**else** **if**(ele[1].equals("ontable"))

ontable[ele[2].charAt(0)%97] =1;

**else** **if**(ele[1].equals("clear"))

clear[ele[2].charAt(0)%97] =1;

**else** **if**(ele[1].equals("hold"))

hold[ele[2].charAt(0)%97] =1;

**else** **if**(ele[1].equals("AE"))

arm =1;

}

}

**public** **int** check(String state)

{

**int** i;

String subG[] = state.split("['^']+");

**int** flag=1;

**for**(i=0;i<subG.length;i++)

{

String ele[] = subG[i].split("[() ]+");

**if**(ele[1].equals("on") &&

on[(**int**)ele[2].charAt(0)%97][(**int**)ele[3].charAt(0)%97] == 1)

flag=1;

**else** **if**(ele[1].equals("ontable") && ontable[ele[2].charAt(0)%97] ==1)

flag =1;

**else** **if**(ele[1].equals("clear") && clear[ele[2].charAt(0)%97] ==1)

flag= 1;

**else** **if**(ele[1].equals("hold") && hold[ele[2].charAt(0)%97] ==1)

flag= 1;

**else** **if**(ele[1].equals("AE") && arm==1)

flag= 1;

**else**

**return** 0;

}

**return** flag;

}

**public** **int** checktop(**char** c)

{

**int** i=0;

**for**(i=0;i<blocks;i++)

{

**if**(on[i][c%97]==1)

**return** i;

}

**return** -1;

}

**public** **void** performAction(String act)

{

String ele[] = act.split("[() ]+");

**if**(act.contains("pick"))

{

ontable[ele[2].charAt(0)%97]=0;

clear[ele[2].charAt(0)%97]=0;

hold[ele[2].charAt(0)%97]=0;

arm=0;

}

**else** **if**(act.contains("unstack"))

{

hold[ele[2].charAt(0)%97]=1;

clear[ele[2].charAt(0)%97]=0;

clear[ele[3].charAt(0)%97]=1;

on[ele[2].charAt(0)%97][ele[3].charAt(0)%97]=0;

arm=0;

}

**else** **if**(act.contains("release"))

{

ontable[ele[2].charAt(0)%97]=1;

clear[ele[2].charAt(0)%97]=1;

hold[ele[2].charAt(0)%97]=0;

arm=1;

}

**else** **if**(act.contains("stack"))

{

hold[ele[2].charAt(0)%97]=0;

clear[ele[2].charAt(0)%97]=1;

clear[ele[3].charAt(0)%97]=0;

on[ele[2].charAt(0)%97][ele[3].charAt(0)%97]=1;

arm=1;

}

}

}

**Planner.java**

import java.util.ArrayList;

import java.util.Scanner;

import java.util.Stack;

public class Planner {

int blocks;

String goal\_s;

State initial,goal,curr;

Stack s;

ArrayList<String> steps;

Planner(int b,String init,String last)

{

blocks = b;

initial = new State(blocks,init);

goal = new State(blocks,last);

curr = new State(blocks,init);

s = new Stack();

goal\_s = last;

steps = new ArrayList<String>();

}

void stackPlan()

{

int i;

String g;

s.push(goal\_s);

String subG[] = goal\_s.split("['^']+");

for(i=subG.length-1;i>=0;i--)

s.push(subG[i]);

while(!s.isEmpty())

{

System.out.println(s);

g = (String)s.pop();

if(g.contains("^"))

{

if(curr.check(g)==0)

{

subG = g.split("['^']+");

for(i=subG.length-1;i>=0;i--)

s.push(subG[i]);

}

}

else if(g.contains("on") && curr.check(g)==0)

{

String ele[] = g.split("[() ]+");

s.push("(stack " + ele[2].charAt(0) + " " + ele[3].charAt(0) +")");

s.push("(clear "+ ele[2].charAt(0) +")^(clear "+ ele[3].charAt(0) +")^"+"(AE)");

s.push("(AE)");

s.push("(clear "+ ele[3].charAt(0) +")");

s.push("(clear "+ ele[2].charAt(0) +")");

}

else if(g.contains("ontable") && curr.check(g)==0)

{

String ele[] = g.split("[() ]+");

s.push("(release " + ele[2].charAt(0) + ")");

s.push("(hold "+ ele[2].charAt(0) +")");

}

else if(g.contains("clear") && curr.check(g)==0)

{

String ele[] = g.split("[() ]+");

if(curr.hold[ele[2].charAt(0)%97]==1 )

{

s.push("(release " + ele[2].charAt(0) +")");

s.push("(hold " + ele[2].charAt(0) +")");

}

else

{

int t =curr.checktop(ele[2].charAt(0));

if(t!=-1)

{

s.push("(unstack "+ Character.toString((char)(t+97))+" " + ele[2].charAt(0) +")");

s.push("(on "+ Character.toString((char)(t+97))+" " + ele[2].charAt(0)

+")^"+"(clear "+ Character.toString((char)(t+97)) +")^"+"(AE)");

s.push("(AE)");

s.push("(clear "+ Character.toString((char)(t+97)) +")");

s.push("(on "+ Character.toString((char)(t+97))+" " + ele[2].charAt(0) +")");

}

}

}

else if(g.contains("hold") && curr.check(g)==0)

{

String ele[] = g.split("[() ]+");

if(curr.ontable[ele[2].charAt(0)%97]==1)

{

s.push("(pick "+ ele[2].charAt(0) +")");

s.push("(ontable " + ele[2].charAt(0) +")^"+"(clear "+ ele[2].charAt(0)

+")^"+"(AE)");

s.push("(AE)");

s.push("(ontable " + ele[2].charAt(0) +")");

s.push("(clear "+ ele[2].charAt(0) +")");

}

else

{

int t =curr.checktop(ele[2].charAt(0));

if(t!=-1)

{

s.push("(unstack "+ Character.toString((char)(t+97))+" " + ele[2].charAt(0) +")");

s.push("(on "+ Character.toString((char)(t+97))+" " + ele[2].charAt(0)

+")^"+"(clear "+ Character.toString((char)(t+97)) +")^"+"(AE)");

s.push("(AE)");

s.push("(clear "+ Character.toString((char)(t+97)) +")");

s.push("(on "+ Character.toString((char)(t+97))+" " + ele[2].charAt(0) +")");

}

}

}

else if(g.contains("AE") && curr.check(g)==0)

{

for(i=0;i<curr.blocks;i++)

{

if(curr.hold[i]==1)

{

s.push("(release " + Character.toString((char)(i+97)) +")");

s.push("(hold " + Character.toString((char)(i+97)) +")");

}

}

}

else if(g.contains("pick") || g.contains("unstack") || g.contains("release") || g.contains("stack"))

{

curr.performAction(g);

steps.add(g);

}

}

printSteps();

}

public void printSteps(){

int i;

System.out.println(""+steps);

}

public static void main(String args[]) {

int b;

String init,goal;

Scanner in = new Scanner(System.in);

System.out.print("Enter number of blocks :");

b=in.nextInt();

in.nextLine();

System.out.print("Enter initial state : ");

init = in.nextLine();

System.out.print("Enter goal state : ");

goal = in.nextLine();

Planner p = new Planner(b, init, goal);

p.stackPlan();

//System.out.println(init + "\n" + goal);

in.close();

}

}