

GoalStack.java

```
package goalstack;
import java.io.*;
import java.util.*;
public class GoalStack
{
    int blocks;
    String goal_string;
    State initial_state,goal_state,current_state;
    Stack s;
    public GoalStack(int blocks,String initial_state,String goal_state)
    {
        this.blocks=blocks;
        this.initial_state=new State(blocks,initial_state);
        this.goal_state=new State(blocks,goal_state);
        this.current_state=new State(blocks,initial_state);
        s=new Stack();
        goal_string=goal_state;
    }
    void stackplan()
    {
        System.out.println("Initial State is: ");
        initial_state.printmatrix();
        System.out.println("Goal State is: ");
        goal_state.printmatrix();
        int location=0;
        s.push(goal_string);
        String split_goal_string[]=goal_string.split("\\^");
        for(int i=split_goal_string.length-1;i>=0;i--)
        {
            s.push(split_goal_string[i]);
        }
        while(!s.isEmpty())
        {
            String pop_element=(String)s.pop();
            if(pop_element.contains("^"))
            {
                System.out.println("-----");
                String pop_goal_split[]=pop_element.split("\\^");
                for(int i=pop_goal_split.length-1;i>=0;i--)
                {
                    s.push(pop_goal_split[i]);
                }
            }
            continue;
        }
    }
}
```

```

StringTokenizer st_pop_element=new StringTokenizer(pop_element,",");
String split_pop_element[]=new String[st_pop_element.countTokens()];
int k=0;
while(st_pop_element.hasMoreTokens())
{
    split_pop_element[k]=st_pop_element.nextToken();
    k++;
}
for(int i=0;i<split_pop_element.length;i++)
    System.out.print(split_pop_element[i]+" ");
System.out.println();
if(split_pop_element[0].equals("on") &&
current_state.on[(int)split_pop_element[1].charAt(0)%97][(int)split_pop_element[2].charAt(0)%97]
==0)
{

    s.push("stack("+split_pop_element[1]+","+split_pop_element[2]+")");
    s.push("hold("+split_pop_element[1]+")");
    s.push("clear("+split_pop_element[2]+")");
}
else if(pop_element.contains("clear") &&
current_state.clear[(int)split_pop_element[1].charAt(0)%97]==0)
{
    location=0;
    for(int i=0;i<blocks;i++)
    {

        if(current_state.on[i][(int)split_pop_element[1].charAt(0)%97]==1)
        {
            location=i;
        }
    }
    s.push("unstack("+(char)(location+97)+","+split_pop_element[1]+")");
    s.push("AE");
    s.push("on("+(char)(location+97)+","+split_pop_element[1]+")");
    s.push("clear("+(char)(location+97)+")");
}
else if(pop_element.contains("AE") && current_state.arm==0)
{
    for(int i=0;i<blocks;i++)
    {
        if(current_state.hold[i]==1)
        {
            location=i;
        }
    }
}

```

```

        }
        s.push("putdown("+(char)(location+97)+"");
        s.push("hold("+(char)(location+97)+"");
    }
    else if(pop_element.contains("ontable") &&
current_state.ontable[(int)split_pop_element[1].charAt(0)%97]==0)
    {
        s.push("putdown("+split_pop_element[1]+"");
        s.push("hold("+split_pop_element[1]+"");
    }
    else if(pop_element.contains("hold") &&
current_state.hold[(int)split_pop_element[1].charAt(0)%97]==0)
    {
        s.push("pickup("+split_pop_element[1]+"");
        s.push("AE");
        s.push("ontable("+split_pop_element[1]+"");
        s.push("clear("+split_pop_element[1]+"");
    }
    else if(split_pop_element[0].equals("stack") ||
split_pop_element[0].equals("putdown") || split_pop_element[0].equals("unstack") ||
split_pop_element[0].equals("pickup"))
    {
        current_state.performaction(split_pop_element);
    }
}
System.out.println("After Goal Stack Planning goal state is: ");
current_state.printmatrix();
}
public static void main(String[] args)throws IOException
{
    BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
    System.out.println("Enter no of blocks: ");
    int blocks=Integer.parseInt(br.readLine());
    System.out.println("Enter initial state: ");
    String initial_state=br.readLine();
    System.out.println("Enter goal state: ");
    String goal_state=br.readLine();
    GoalStack obj=new GoalStack(blocks,initial_state,goal_state);
    obj.stackplan();
}
}

```

State.java

```

package goalstack;
import java.util.StringTokenizer;

```

```

public class State
{
    int blocks;
    int on[][];
    int ontable[];
    int clear[];
    int hold[];
    int arm;
    State(int blocks,String state_string)
    {
        this.blocks=blocks;
        on=new int[blocks][blocks];
        ontable=new int[blocks];
        clear=new int[blocks];
        hold=new int[blocks];
        arm=-1;
        setstate(state_string);
    }
    void setstate(String state_string)
    {
        //System.out.println(state_string);
        StringTokenizer st=new StringTokenizer(state_string,"^");
        String op[]=new String[st.countTokens()];
        int k=0;
        while(st.hasMoreTokens())
        {
            op[k]=st.nextToken();
            k++;
        }
        for(int i=0;i<k;i++)
        {
            StringTokenizer st_op=new StringTokenizer(op[i],",");
            int l=0;
            String op1[]=new String[st_op.countTokens()];
            while(st_op.hasMoreTokens())
            {
                op1[l]=st_op.nextToken();
                l++;
            }
            //System.out.println(op1[0]);
            if(op1[0].equals("on"))
            {
                on[(int)op1[1].charAt(0)%97][(int)op1[2].charAt(0)%97]=1;
            }
        }
    }
}

```

```

        else if(op1[0].equals("ontable"))
        {
            ontable[(int)op1[1].charAt(0)%97]=1;
        }
        else if(op1[0].equals("clear"))
        {
            clear[(int)op1[1].charAt(0)%97]=1;
        }
        else if(op1[0].equals("hold"))
        {
            hold[(int)op1[1].charAt(0)%97]=1;
        }
        else if(op1[0].equals("AE"))
        {
            arm=1;
        }
    }
}

void performaction(String[] split_pop_element)
{
    if(split_pop_element[0].equals("stack"))//done
    {

        on[(int)split_pop_element[1].charAt(0)%97][(int)split_pop_element[2].charAt(0)%97]=1;
        clear[(int)split_pop_element[1].charAt(0)%97]=1;
        clear[(int)split_pop_element[2].charAt(0)%97]=0;
        hold[(int)split_pop_element[1].charAt(0)%97]=0;
        arm=1;
    }
    else if(split_pop_element[0].equals("unstack"))//done
    {

        on[(int)split_pop_element[1].charAt(0)%97][(int)split_pop_element[2].charAt(0)%97]=0;
        clear[(int)split_pop_element[1].charAt(0)%97]=0;
        clear[(int)split_pop_element[2].charAt(0)%97]=1;
        hold[(int)split_pop_element[1].charAt(0)%97]=1;
        arm=0;
    }
    else if(split_pop_element[0].equals("putdown"))//done
    {
        ontable[(int)split_pop_element[1].charAt(0)%97]=1;
        clear[(int)split_pop_element[1].charAt(0)%97]=1;
        hold[(int)split_pop_element[1].charAt(0)%97]=0;
        arm=1;
    }
}

```

```

else if(split_pop_element[0].equals("pickup"))//done
{
    hold[(int)split_pop_element[1].charAt(0)%97]=1;
    clear[(int)split_pop_element[1].charAt(0)%97]=0;
    ontable[(int)split_pop_element[1].charAt(0)%97]=0;
    arm=0;
}
}
void printmatrix()
{
    System.out.println("ON Matrix");
    for(int i=0;i<blocks;i++)
    {
        for(int j=0;j<blocks;j++)
        {
            System.out.print(on[i][j]+" ");
        }
        System.out.println();
    }
    System.out.println("Ontable");
    for(int i=0;i<blocks;i++)
    {
        System.out.print(ontable[i]+" ");
    }
    System.out.println();
    System.out.println("clear");
    for(int i=0;i<blocks;i++)
    {
        System.out.print(clear[i]+" ");
    }
    System.out.println();
    System.out.println("hold");
    for(int i=0;i<blocks;i++)
    {
        System.out.print(hold[i]+" ");
    }
    System.out.println();
}
}

```

Output –

Enter no of blocks:

4

Enter initial state:

on(b,a)^ontable(c)^ontable(a)^ontable(d)^clear(c)^clear(d)^clear(b)^AE

Enter goal state:

on(c,a)^on(b,d)^ontable(a)^ontable(d)^clear(c)^clear(b)^AE

Initial State is:

ON Matrix

0 0 0 0

1 0 0 0

0 0 0 0

0 0 0 0

Ontable

1 0 1 1

clear

0 1 1 1

hold

0 0 0 0

Goal State is:

ON Matrix

0 0 0 0

0 0 0 1

1 0 0 0

0 0 0 0

Ontable

1 0 0 1

clear

0 1 1 0

hold

0 0 0 0

After Goal Stack Planning goal state is:

ON Matrix

0 0 0 0

0 0 0 1

1 0 0 0

0 0 0 0

Ontable

1 0 0 1

clear

0 1 1 0

hold

0 0 0 0