using System;

using System.Collections.Generic;

namespace WaterJug

{

public class Program

{

int c1 = 0;

int c2 = 0;

int c3 = 0;

public Program(int c1, int c2)

{

this.c1 = c1;

this.c2 = c2;

this.c3 = 8 - c1 - c2;

}

public bool IsGoal

{

get

{

return (c2 == 4 && c3 == 4);

}

}

public override bool Equals(Object ob)

{

Program st = (Program)ob;

if (this.c1 == st.c1 && this.c2 == st.c2 && this.c3 == st.c3)

{

return true;

}

else

{

return false;

}

}

public List<Program> getChildren()

{

List<Program> children = new List<Program>();

//from c3 to c1

if (c3 != 0 && c1 != 3)

{

if (c1 + c3 <= 3)

{

children.Add(new Program(c1 + c3, 8 - c3 - c1));

}

else

{

children.Add(new Program(3, 8 - c3 - c1));

}

}

//from c1 to c2

if (c1 != 0 && c2 != 5)

{

if (c1 + c2 <= 5)

{

children.Add(new Program(0, c1 + c2));

}

else

{

children.Add(new Program(c1 + c2 - 5, 5));

}

}

//from c1 to c3

if (c1 != 0 && c3 != 8)

{

children.Add(new Program(0, 8 - c1 - c3));

}

//from c2 to c3

if (c2 != 0 && c1 != 3)

{

if (c1 + c2 <= 3)

{

children.Add(new Program(c1 + c2, 0));

}

else

{

children.Add(new Program(3, c1 + c2 - 3));

}

}

//from c2 to c3

if (c2 != 0 && c3 != 8)

{

children.Add(new Program(8 - c2 - c3, 0));

}

//from c3 to c2

if (c3 != 0 && c2 != 5)

{

if (c3 + c2 <= 5)

{

children.Add(new Program(8 - c3 - c2, c3 + c2));

}

else

{

children.Add(new Program(8 - c3 - c2, 5));

}

}

return children;

}

public override string ToString()

{

return "{" + c1 + "," + c2 + "," + c3 + "}";

}

}

}

using System;

using System.Collections.Generic;

using System.Text;

namespace WaterJug

{

class JugBFS

{

private static List<Program> visitState = new List<Program>();

private static Queue<Program> stateQueue = new Queue<Program>();

public static void Main()

{

Program currentState = new Program(0, 0);

stateQueue.Enqueue(currentState);

while (stateQueue.Count != 0)

{

Program firstIQueue = stateQueue.Peek();

if (firstIQueue.IsGoal)

{

foreach (Program state in visitState)

{

Console.WriteLine(state.ToString());

}

return;

}

else

{

visitState.Add(firstIQueue);

List<Program> children = firstIQueue.getChildren();

foreach (Program state in children)

{

if (!visitState.Contains(state))

{

stateQueue.Enqueue(state);

}

}

stateQueue.Dequeue();

}

}

}

}

}