public static void my\_by\_bridge(List<my\_bridge> BridgeValue, int bridge\_num)

{

int[,] cross\_matrix = new int[100, 100];

int[] cross\_num = { 0 };

Dictionary<int, int> ValueBridge = new Dictionary<int, int>();

List<int> Values = new List<int>();

int i, j;

bool flag = false;

for (i = 0; i < BridgeValue.Count; i++)

{

for (j = i + 1; j < BridgeValue.Count; j++)

{

if (my\_check\_intersect(BridgeValue[i], BridgeValue[j]))

{

//cross\_matrix[i,j] = 1;

//cross\_matrix[j,i] = 1;

//cross\_num[i]++;

//cross\_num[j]++;

Console.WriteLine(BridgeValue[i].id + ":::: Intersects:::::" + BridgeValue[j].id );

flag = true;

if (ValueBridge.ContainsKey(i))

{

ValueBridge[i] = ValueBridge[i] + 1;

}

else

{

ValueBridge.Add(i, 1);

}

}

}

if(!flag)

{

Values.Add(i+1);

}

flag = false;

}

Console.WriteLine("hi");

//while ((i = my\_getMax(cross\_num, bridge\_num)) >= 0)

//{

// BridgeValue[i].id \*= -1;

// cross\_num[i] \*= -1;

// for (j = 0; j < bridge\_num; j++)

// {

// if (BridgeValue[j].id < 0)

// continue;

// if (cross\_matrix[i][j]==1)

// cross\_num[j]--;

// }

//}

}

public static int my\_getMax(int[] cross\_num, int bridge\_num)

{

int i, max = 0, index = -1;

for (i = 0; i<bridge\_num; i++)

{

if (cross\_num[i] > max)

{

max = cross\_num[i];

index = i;

}

}

return index;

}

my\_by\_bridge(BridgeValues, BridgeValues.Count);

//for (int i = 0; i < BridgeValues.Count; i++)

//{

// if (BridgeValues[i].id > 0)

// Console.WriteLine(BridgeValues[i].id);

//}