public class Card : ICard;

{

public int Number;

public Suite suite;

public IsBigJoke;

public IsSmallJoke;

}

public enum Suite{

Diamond,

Heart,

Spade,

Club,

}

public class Player : IPayer

{

public Card[] handOfCards;

public List<Card> play();

}

public class PlayEngine : IPlayEngine

{

public IPlayer[] Play();

}

public class CardGame{

public CardGame(int totalCards);

public CardGame(ICard[] cards);

private ICard[] cards;

private IPlayer[] allplayers[];

private IPlayer[] winer;

private IPlayEngine playEngine;

public ICard[] Card

{

get;set;

}

}

public Interface ICard;

public Interface IPlayer;

public Interface IplayEngine;

--------------------------------------------------

Tests:

- what you are testing, sample, sample output

[], 10, --- null;

[1], 20 --- null;

[1,2],3 --- [ [1,2] ]

[1,2,3,4,5], 1000 --- []

[2,3,1,7,4,5], 9 ------ [ [2,7], [4,5] ]

[2,3,1,7,4,5,2,3,1,7,4,5,2,3,1,7,4,5,2,3,1,7,-4,5,2,3,1,7,4,5], 9 ---- [[2,7], [4,5]]

[-2,3,1,-7,4,5], -9 ------ [ [-2,-7] ]

[1,1,1,1,1,1,1,1,1,1,1,1,1,1] , 2 ---------- [1,1]

public HashSet<List<int>> FindPairSum(int[] input, int vlaue)

{

if(input.Length< 2) return null;

HashSet<List<int>> res = new HashSet<List<int>>();

HashSet<int> hashSet = new HashSet<int>();

for(int i = 0;i< input.Length;i++)

{

if(hashSet.Contains(vlaue-input[i])

{

List<int> tmp = new List<int>() {value-input[i], input[i]};

tmp.Sort();

if(!res.Contains(tmp)) // assumme this will eliminate the duplicates.

{

res.add(tmp);

}

}

if(!hashSet.Contains(input[i])

{

hashSet.add(input[i]);

}

}

return res;

}