public class Solution {

public IList<string> GenerateParenthesis(int n)

{

IList<string> combinations = new List<string>();

generateAll(new char[2 \* n], 0, combinations);

return combinations;

}

public virtual void generateAll(char[] current, int pos, IList<string> result)

{

if (pos == current.Length)

{

if (valid(current))

{

result.Add(new string(current));

}

}

else

{

current[pos] = '(';

generateAll(current, pos + 1, result);

current[pos] = ')';

generateAll(current, pos + 1, result);

}

}

public virtual bool valid(char[] current)

{

int balance = 0;

foreach (char c in current)

{

if (c == '(')

{

balance++;

}

else

{

balance--;

}

if (balance < 0)

{

return false;

}

}

return (balance == 0);

}

}

Detyra 2