```
using System;

namespace Unit4
{
    class StackUtils
    {
        public static Stack<T> CreateStackFromArray<T>(T[] arr)
        {
            Stack<T> s = new Stack<T>();

            return s;
        }
        public static Stack<T> CreateStackFromArrayEnd<T>(T[] arr)
        {
            Stack<T> s = new Stack<T>();

            return s;
        }

        public static void SpilledOn<T>(Stack<T> to, Stack<T> from)
        {
        }
        public static Stack<T> Clone<T>(Stack<T> s)
        {
            Stack<T> t = new Stack<T>();
            Stack<T> s2 = new Stack<T>();
            return s2;
        }
        public static Stack<T> CloneRec<T>(Stack<T> s)
        {
            return new Stack<T>();
        }
        public static Stack<T> ReverseStack<T>(Stack<T> s)
        {
            Stack<T> t = new Stack<T>();
            Stack<T> t2 = new Stack<T>();
            return t2;
        }
        public static int GetSize<T>(Stack<T> s)
        {
            return 0;
        }
        public static int GetSizeRec<T>(Stack<T> s)
        {
            return 0;
        }
        public static int Sum(Stack<int> s)
        {
            return 0;
```

```
        }
        public static int SumRec(Stack<int> s)
        {
            return 0;
        }
        public static bool IsSorted(Stack<int> s)
        {
            return false;
        }
        public static void Sort(Stack<int> s)
        {
        }

        public static bool IsSortedRec(Stack<int> s)
        {
            return false;
        }
        public static bool IsExist(Stack<int> s, int val)
        {
            return false;
        }
        public static bool IsExistRec(Stack<int> s, int val)
        {
            return false;
        }
    }
}

```