A stack class for characters

```
/*
C#: The Complete Reference
by Herbert Schildt

Publisher: Osborne/McGraw-Hill (March 8, 2002)
ISBN: 0072134852
*/

// A stack class for characters.

using System;

class Stack {
// these members are private
char[] stck; // holds the stack
int tos; // index of the top of the stack

// Construct an empty Stack given its size.
public Stack(int size) {
stck = new char[size]; // allocate memory for stack
tos = 0;
}

// Construct a Stack from a stack.
public Stack(Stack ob) {
// allocate memory for stack
stck = new char[ob.stck.Length];

// copy elements to new stack
for(int i=0; i < ob.tos; i++) stck[i] = ob.stck[i]; // set tos
for new stack tos = ob.tos; } // Push characters onto the
stack. public void push(char ch) { if(tos==stck.Length) {
Console.WriteLine(" -- Stack is full."); return; } stck[tos] =
ch; tos++; } // Pop a character from the stack. public char
pop() { if(tos==0) { Console.WriteLine(" -- Stack is empty.");
return (char) 0; } tos--; return stck[tos]; } // Return true
```

if the stack is full. public bool full() { return tos==stck.Length; } // Return true if the stack is empty. public bool empty() { return tos==0; } // Return total capacity of the stack. public int capacity() { return stck.Length; } // Return number of objects currently on the stack. public int getNum() { return tos; } } // Demonstrate the Stack class. public class StackDemo1 { public static void Main() { Stack stk1 = new Stack(10); char ch; int i; // Put some characters into stk1. Console.WriteLine("Push A through Z onto stk1."); for(i=0; !stk1.full(); i++) stk1.push((char) ('A' + i)); // Create a copy of stck1 Stack stk2 = new Stack(stk1); // Display the contents of stk1. Console.Write("Contents of stk1: "); while(!stk1.empty()) { ch = stk1.pop(); Console.Write(ch); } Console.WriteLine(); Console.Write("Contents of stk2: "); while (!stk2.empty()) { ch = stk2.pop(); Console.Write(ch); } Console.WriteLine(" "); } } [/csharp]