using System;

class Stack

{

private int[] data;

private int top;

// Constructor to initialize the stack

public Stack(int size)

{

data = new int[size];

top = -1;

}

// Property to check if the stack is empty

public bool IsEmpty

{

get { return top == -1; }

}

// Property to check if the stack is full

public bool IsFull

{

get { return top == data.Length - 1; }

}

// Method to push an element onto the stack

public void Push(int value)

{

if (IsFull)

{

Console.WriteLine("Stack overflow. Cannot push element onto the stack.");

}

else

{

top++;

data[top] = value;

Console.WriteLine($"Pushed {value} onto the stack.");

}

}

// Method to pop an element from the stack

public int Pop()

{

if (IsEmpty)

{

Console.WriteLine("Stack underflow. Cannot pop element from an empty stack.");

return -1; // Assuming -1 represents an error or an invalid value

}

else

{

int poppedValue = data[top];

top--;

Console.WriteLine($"Popped {poppedValue} from the stack.");

return poppedValue;

}

}

}

class Program

{

static void Main()

{

Console.Write("Enter the size of the stack: ");

int size = int.Parse(Console.ReadLine());

Stack stack = new Stack(size);

stack.Push(10);

stack.Push(20);

stack.Push(30);

int poppedValue = stack.Pop();

Console.WriteLine($"Popped value: {poppedValue}");

stack.Push(40);

Console.ReadLine(); // To keep the console window open

}

}