

Assume the existence of a `StackOfInt` class, the relevant methods of which are these:

```
// Reports whether or not the stack is empty
public boolean isEmpty() { ... }

// Returns the item at the top of the stack
public int topOf() { ... }

// Places the given item on top of the stack
public void push(int k) { ... }

// Removes the item at the top of the stack and returns it
public int pop() { ... }
```

Supply the missing body of the method below so that it fulfills its intended purpose, as described in the comments preceding its heading. Any auxiliary storage structure employed must itself be a stack. (That is, you are not to make use of an array, `ArrayList`, or `String`.) The first two lines are provided.

```
/* Returns a new stack that is identical to the given one.
** The given stack must be restored to its original state.
** For example, if the items on the given stack were A, B, C, and D
** (top to bottom), then the new stack returned by this method will
** contain A, B, C, and D (top to bottom), as will the given stack.
*/
public static StackOfInt cloneOf(StackOfInt stk) {
    StackOfInt result = new StackOfInt(); // stack to be returned
    StackOfInt auxStk = new StackOfInt(); // auxiliary stack
```