Table 1

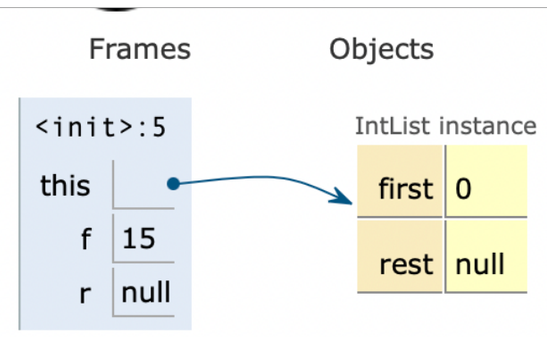
Pointers

recursion code

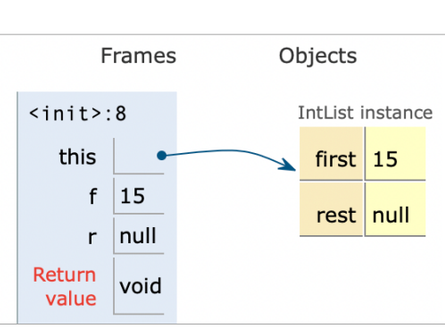
```
public static IntList incrList(IntList L, int x) {
        if (L == null) {
            return L;
        }
        IntList incr = new IntList(L.first + x, incrList(L.rest, x));
        return incr;
    }
```

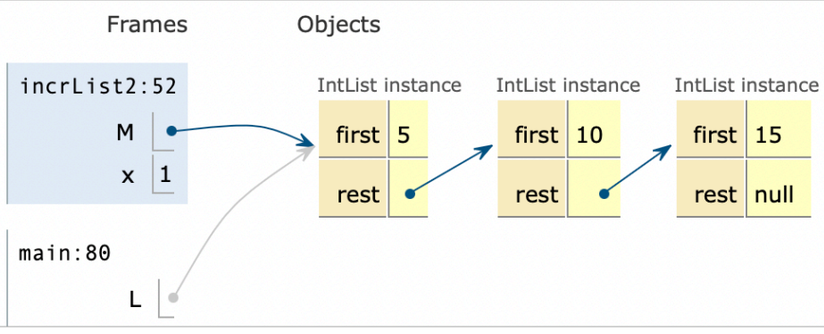
CS61B lists1/exercises/ExtraIntListPractice.java - iterative method

	new creates a box with 64bits- address in it, which points to the object with default value; When we **instantiate** an Object (e.g. Dog, Walrus, Planet): Java first allocates a box of bits for each instance variable of the class and fills them with a default value (e.g. 0, null). parameter passing: copy the bits into the new scope

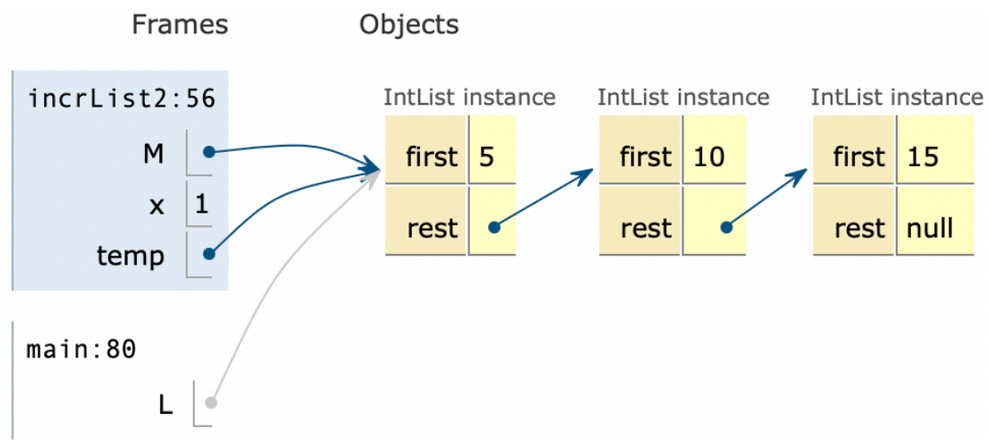
`public static void main(String[] args) {`
`  IntList L = new IntList(15, null);`
`  L = new IntList(10, L);`
`  L = new IntList(5, L);`
	

constructor	after going through the constructor: The constructor then usually fills every such box with some other value.

`public IntList(int f, IntList r) {`
`  first = f;`
`  rest = r;`
`}`
	

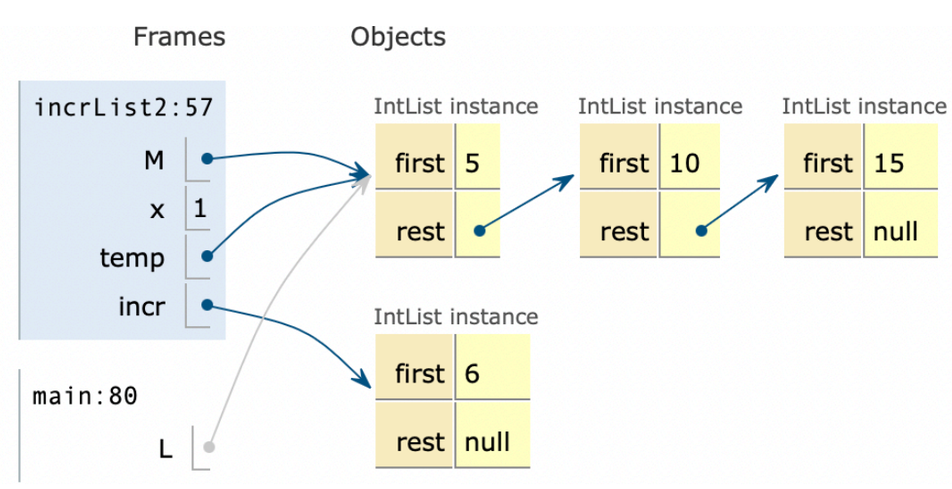
`public static void main(String[] args) {`
`  IntList L = new IntList(15, null);`
`  L = new IntList(10, L);`
`  L = new IntList(5, L);`
 	
`  IntList N = IntList.incrList2(L, 1);`
`  System.out.println(L.get_Recur(0));`
	

	pointer temp

`public static IntList incrList2(IntList M, int x)`
`  if (M == null) {`
`  return M;`
`  }`
`  IntList temp = M;`
`  IntList incr = new IntList(temp.first + x, nu`
`  IntList temp2 = incr;`
`  while (temp.rest != null) {`
`  temp = temp.rest;`
`  temp2.rest = new IntList(temp.first + x,`
`  temp2 = temp2.rest;`
`  }`
`  return incr;`
`}`
	

创建new IntList object	

```
    public static IntList incrList2(IntList M, int x)
        if (M == null) {
            return M;
        }
        IntList temp = M;
        IntList incr = new IntList(temp.first + x, nu
        IntList temp2 = incr;
        while (temp.rest != null) {
            temp = temp.rest;
            temp2.rest = new IntList(temp.first + x,
            temp2 = temp2.rest;
        }
        return incr;
    }
```



第二个pointer temp2

```
    public static IntList incrList2(IntList M, int x)
        if (M == null) {
            return M;
        }
        IntList temp = M;
        IntList incr = new IntList(temp.first + x, nu
        IntList temp2 = incr;
        while (temp.rest != null) {
            temp = temp.rest;
            temp2.rest = new IntList(temp.first + x,
            temp2 = temp2.rest;
        }
        return incr;
    }
```