longest.icn:

The instructor's solution:

```plaintext
procedure main()
    longest := "x"
    lines := ""

    while line := read() || "\n" do {
        if line[*longest || "x") then {
            longest := line
            lines := line
        } else if line[*longest] then
            lines ||:= line
    }

    writes(lines)
end
```

Mr. Linn's solution was perhaps the cleanest and easiest to understand:

```plaintext
procedure main()
    max := total := read() || "\n"

    every temp := !&input || "\n" do
        if temp[*max] & max[*temp] then
            total ||:= temp
        else if temp[*max] then
            total := max := temp

    writes( total )
end
```
seqwords.icn:

The instructor's solution:

```plaintext
procedure main()
    lines := ""
    while (line := read()) ~== "." do
        lines ::= left(line,1000)
    while write(doseq(lines))
end

procedure doseq(lines)
    s ::= ""
    while num := read() do {
        if *num = 0 then next  # ignore empty lines
        if num == "." then break
        s ::= trim(lines[(num-1)*1000+1:+1000]) || " "
    }
    return s[1:-1]
end
```

boxlines.icn:

The instructor's solution:

```plaintext
global max  # lazy...

procedure main()
    max ::= 0
    b ::= box()
    writes(bar())
    writes(b)
end

procedure box()
    if not (line ::= reverse(trim(reverse(trim(read()))))) then
        return ""
    max ::= *line
    b ::= box()
    return "| " || center(line, max) || " |" || bar() || b
end

procedure bar()
    return "|" || repl("-", max+2) || "\n" end
```