Exercise 10: Syntax Error Handling with General Anchors

Write parser methods for the following grammar and implement error handling with general anchors (Section 3.4.2). You can write the set operations in pseudocode.

```
A = { b C | d }.
C = [ c ] e.
```

A is the start symbol, whose successor is eof.

Solution

The successor of A is eof, the possible successors of C are b and d (since a further iteration can follow) as well as eof (if the iteration ends). The parser method of A is invoked as A({eof}).

```
static void A (BitSet sux) {
    for (;;) {
        if (sym == b) {                  // first alternative => stay in loop
            scan();
            C({b, d} ∪ sux);
        } else if (sym == d) {           // second alternative => stay in loop
            scan();
        } else if (sym ∈ sux) {          // leave loop
            break;
        } else {                         // error => stay in loop
            error("b or d expected", {b, d} ∪ sux);
            // recover with b or d or sux
        }
    }
}

static void C (BitSet sux) {
    if (sym != c && sym != e) {
        error("c or e expected", {c, e} ∪ sux);
        // recover with c or e or sux
    }
    if (sym == c) scan();
    check(e, sux);
}
```

The method error() prints an error message and then skips symbols until a symbol from the successor set sux occurs (see Section 3.4.2).

```
static void error (String msg, BitSet sux) {
    if (errDist >= 3) {
        System.out.println("line " + la.line + ", col " + la.col + ": " + msg);
        errors++;
    }
    while (!sux.get(sym)) scan();
    errDist = 0; // restart counting
}
```