Lab 6.1

```
#include <iostream>
#include <stack>
#include <string>

using namespace std;

int main() {
    string s;
    cout << "Enter string: ";
    cin >> s;

    enum State { A_EVEN, A_ODD, IN_B, DEAD };
    State st = A_EVEN;

    stack<char> stk;
    stk.push('$'); // bottom marker

    bool seen_b = false;

    for (size_t i = 0; i < s.length(); i++) {
        char c = s[i];

        if (st == DEAD) break;

        if (st == A_EVEN) {
            if (c == 'a') {
                st = A_ODD;  // first a of the pair
            }
```

```
    else if (c == 'b') {
        // Start reading b's
        seen_b = true;
        st = IN_B;
        if (!stk.empty() && stk.top() == 'X')
            stk.pop();
        else
            st = DEAD; // not enough pairs of a's for this b
    }
    else {
        st = DEAD; // invalid character
    }
}
else if (st == A_ODD) {
    if (c == 'a') {
        // second a of the pair
        stk.push('X');
        st = A_EVEN;
    }
    else {
        st = DEAD; // b starts in odd phase ? invalid
    }
}
else if (st == IN_B) {
    if (c == 'b') {
        seen_b = true;
        if (!stk.empty() && stk.top() == 'X')
            stk.pop();
        else
```

```
            st = DEAD; // too many b's
        }
        else {
            st = DEAD; // no a's allowed after b's
        }
      }
    }

    bool accept =
      (st != DEAD) &&
      seen_b &&                    // must have at least 1 b
      (st == IN_B || st == A_EVEN) && // ended in valid state
      stk.size() == 1 && stk.top() == '$'; // stack back to bottom marker

    if (accept)
      cout << "Accepted\n";
    else
      cout << "Rejected\n";

    return 0;
}
```

Lab 6.2

```
#include <iostream>
#include <stack>
#include <string>

using namespace std;

int main() {
    string s;
    cout << "Enter string: ";
    cin >> s;

    enum State { READING_A, FIRST_B, SECOND_B, READING_C, DEAD };
    State st = READING_A;

    stack<char> stk;
    stk.push('$'); // bottom marker

    bool seen_b = false;
    bool seen_c = false;

    for (size_t i = 0; i < s.length(); i++) {
        char c = s[i];

        if (st == DEAD) break;

        if (st == READING_A) {
            if (c == 'a') {
                stk.push('X');
```

```
    }
    else if (c == 'b') {
        if (stk.size() > 1) {
            seen_b = true;
            st = FIRST_B;
        } else {
            st = DEAD; // b's without a's
        }
    }
    else {
        st = DEAD;
    }
}
else if (st == FIRST_B) {
    if (c == 'b') {
        st = SECOND_B;
    } else {
        st = DEAD; // must have 2 b's per a
    }
}
else if (st == SECOND_B) {
    if (c == 'b') {
        // Still expecting another b for the next a
        st = FIRST_B;
    }
    else if (c == 'c') {
        seen_c = true;
        if (stk.top() == 'X') {
            stk.pop(); // matched one a with its 2 b's and 1 c
```

```
            st = READING_C;
        } else {
            st = DEAD;
        }
    }
    else {
        st = DEAD;
    }
  }
  else if (st == READING_C) {
    if (c == 'c') {
        seen_c = true;
        if (stk.top() == 'X') {
            stk.pop();
        } else {
            st = DEAD;
        }
    }
    else {
        st = DEAD; // no b's after c's allowed
    }
  }
}

bool accept =
  (st != DEAD) &&
  seen_b && seen_c &&
  stk.size() == 1 && stk.top() == '$';
```

```
    if (accept)
        cout << "Accepted\n";
    else
        cout << "Rejected\n";

    return 0;
}
```

Lab 6.3

```
#include <iostream>
#include <stack>
#include <string>

using namespace std;

int main() {
    string s;
    cout << "Enter string: ";
    cin >> s;

    enum State { READING_A, READING_B, DEAD };
    State st = READING_A;

    stack<char> stk;
    stk.push('$'); // bottom marker

    bool seen_b = false;
```

```
for (size_t i = 0; i < s.length(); i++) {
    char c = s[i];

    if (st == DEAD) break;

    if (st == READING_A) {
        if (c == 'a') {
            stk.push('X');
        }
        else if (c == 'b') {
            if (stk.top() == 'X') {
                stk.pop();
                st = READING_B;
                seen_b = true;
            } else {
                st = DEAD; // b's without matching a's
            }
        }
        else {
            st = DEAD; // invalid symbol
        }
    }
    else if (st == READING_B) {
        if (c == 'b') {
            if (stk.top() == 'X') {
                stk.pop();
                seen_b = true;
            } else {
                st = DEAD; // too many b's
```

```
            }
        }
        else {
            st = DEAD; // no a's allowed after b's
        }
      }
    }

    bool accept =
      (st != DEAD) &&
      seen_b &&
      stk.size() == 1 && stk.top() == '$';

    if (accept)
      cout << "Accepted\n";
    else
      cout << "Rejected\n";

    return 0;
}
```