```
#include <iostream>
#include <stdio.h>
#include <stdlib.h>

using namespace std;

bool above_threshold(const char *line, int column, int threshold) {
    char buffer[20];
    int pos = 0;
    int i = 0;

    while (column && line[pos]) {
        if (line[pos] != ' ' && (pos == 0 || line[pos-1] == ' ')) {
            --column;
        }
        ++pos;
    }

    if (column) {
        cout << "Invalid Column" << endl;
        exit(1);
    }

    --pos;
    while(line[pos] != ' ' && line[pos] != '\n') {
        buffer[i] = line[pos];
        ++i;
        ++pos;
    }

    buffer[i] = '\0';

    return (atoi(buffer) > threshold);
}

// assume valid lines only have space or digits and are non-empty
bool valid_line(const char *line) {
    int pos = 0;
    while(line[pos] != '\n') {
        if (line[pos] != ' ' && (line[pos] < '0' || line[pos] > '9')) {
            return false;
        }
        ++pos;
    }
    return pos;
}
```

```
47
48 int main(int argc, const char *argv[]) {
49     char line[100];
50     int cur_n = 0;
51
52     int column = atoi(argv[1]);
53     int threshold = atoi(argv[2]);
54     int n = atoi(argv[3]);
55
56     while(fgets(line, 100, stdin) != NULL) {
57         if (valid_line(line) && above_threshold(line, column, threshold)) {
58             ++cur_n;
59             if (cur_n == n) {
60                 // only report n consecutive times. keep quiet for > n
61                 cout << "Column " << column << " above threshold "
62                      << threshold << " " << cur_n << " times." << endl;
63             }
64         } else {
65             cur_n = 0;
66         }
67     }
68     return 0;
69 }
```