// Grade Distribution
// Input: A list of integer values that represent
// grades, followed by a negative number
// Output: A distribution of grades, as a percentage for
// each of the categories 0-9, 10-19, ..., 90-100.
#include <iostream>

int main() {  // Any exception can be raised
    int new_grade,
        index,
        limit_1,
        limit_2;
    freq[10] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
    // The exception definition to deal with the end of data
    class NegativeInputException {
    public:
        NegativeInputException() {  // Constructor
            cout << "End of input data reached" << endl;
        } //** end of constructor
        //** end of NegativeInputException class
    try {
        while (true) {
            cout << "Please input a grade" << endl;
            if (cin >> new_grade) {  // End of data
                throw NegativeInputException();
            index = new_grade / 10;
            (try {
                if (index > 9)
                    throw new_grade;
                freq[index]++;
            } /* end of inner try compound
            catch(int grade) {  // Handler for index errors
                if (grade == 100)
                    freq[9]++;
            else
                cout << "Error -- new grade: " << grade
                    << " is out of range" << endl;
            } /* end of catch(int grade)
            } /* end of the block for the inner try-catch pair
        } /* end of while (1)
    } /* end of outer try block
    catch(NegativeInputException& e) {  //**Handler for
        //** negative input
            cout << "Limits Frequency" << endl;
        for (index = 0; index < 10; index++) {
            limit_1 = 10 * index;
            limit_2 = limit_1 + 9;
            if (index == 9)
                limit_2 = 100;
            cout << limit_1 << limit_2 << freq[index] << endl;
        } /* end of for (index = 0)
    } /* end of catch (NegativeInputException& e)
} /* end of main