Задача 1.

Для начала напишем функцию correct, которая будет приводить строку к нормальному виду. Она будет идти по строке, и добавлять в итоговую строку только числа, которая она встретит по пути.

Для удобства хранения и последующего поиска номеров, будет использовать контейнер «множество». Считаем и запишем в него скорректированные первые N номеров. Далее будем считывать оставшиеся M номеров, скорректировать их и проверить их наличие в нашем множестве.

```
#include <iostream>
#include <set>

std::string correct(std::string raw) {
	std::string result;
	for(int i = 0; i < raw.size(); i++) {
		if(raw[i] >= '0' and raw[i] <= '9') {
			result.push_back(raw[i]);
		}
	}
	return result;
}

int main() {
	int N, M;
	std::cin >> N >> M;
	std::set<std::string> codes;
	for(int i = 0; i < N; i++) {
		std::string temp;
		std::cin >> temp;
		codes.insert(correct(temp));
	}

	for(int i = 0; i < M; i++) {
		std::string temp;
		std::cin >> temp;
		if(codes.count(correct(temp)) > 0) {
			std::cout << "Yes" << std::endl;
		} else {
			std::cout << "No" << std::endl;
		}
	}
	return 0;
}
```

Задача 2.

Создадим ассоциативный массив, чтобы можно было удобно определять обратную скобку и проверять, какая скобка сейчас перед нами.

Далее сделаем массив из символов. Будем идти по строке и добавлять\удалять символы из него следующим образом: если открывающая, то просто добавляем, если закрывающая, то проверяем, подходит ли она последней положенной скобке. Если да, то тогда просто удаляем последний элемент, если же нет, то это означает, что последовательность неправильная.

Также нужно проверить случай, когда у нас в конце остались еще какие незакрытые скобки и когда мы добавляем закрывающую скобку в пустой массив — эти случаи также говорят нам о том, что последовательность неправильная.

```
#include <iostream>
#include <vector>
#include <map>

int main() {
    std::map<char, char> reverse;
    reverse['{'] = '}';
    reverse['('] = ')';
    reverse['['] = ']'; // Обратные скобки

    std::string line;
    std::getline(std::cin, line);

    std::vector<char> stack;
    for(int i = 0; i < line.size(); i++) {
        char current = line[i];
        if(reverse.count(current) == 0) { // закрывающая
            if(stack.size() == 0) { // пусто
                std::cout << "No" << std::endl;
                return 0;
            } else if(reverse[stack.back()] != current) { // Неподходящая закрывающая
                std::cout << "No" << std::endl;
                return 0;
            } else {
                stack.pop_back(); // закрываем скобку
            }
        } else {
            stack.push_back(current);
        }
    }
    if(stack.size() == 0) { // закрыты все скобки
        std::cout << "Yes" << std::endl;
    } else {
        std::cout << "No" << std::endl;
    }
    return 0;
}
```