

Q3. //////////////////////////////////////

```
#include <iostream>
#include <vector>
#include <queue>
#include <string>
#include <unordered_set>
```

```
using namespace std;
```

```
struct Medicine {
    int days;
    int removes;
    int side_effects;
};
```

```
int symptoms_bitmask(const string& symptoms) {
    int mask = 0;
    for (char ch : symptoms) {
        mask = (mask << 1) | (ch - '0');
    }
    return mask;
}
```

```
int bfs_min_days(int n, int initial_symptoms, const vector<Medicine>& medicines) {
    queue<pair<int, int>> q;
    unordered_set<int> visited;

    q.push({initial_symptoms, 0});
    visited.insert(initial_symptoms);
```

```

while (!q.empty()) {
    int current_state = q.front().first;
    int days = q.front().second;
    q.pop();

    if (current_state == 0) {
        return days;
    }

    for (const auto& med : medicines) {
        int next_state = current_state;
        next_state &= ~med.removes; // remove symptoms
        next_state |= med.side_effects; // add side effects

        if (visited.find(next_state) == visited.end()) {
            q.push({next_state, days + med.days});
            visited.insert(next_state);
        }
    }
}

return -1;
}

```

```

int main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);

    int t;

```

```

cin >> t;

while (t--) {
    int n, m;
    cin >> n >> m;

    string initial_symptoms;
    cin >> initial_symptoms;

    int initial_state = symptoms_bitmask(initial_symptoms);

    vector<Medicine> medicines(m);
    for (int i = 0; i < m; ++i) {
        cin >> medicines[i].days;

        string removes, side_effects;
        cin >> removes >> side_effects;

        medicines[i].removes = symptoms_bitmask(removes);
        medicines[i].side_effects = symptoms_bitmask(side_effects);
    }

    int result = bfs_min_days(n, initial_state, medicines);
    cout << result << "\n";
}

return 0;
}

```

```

////////////////////////////////////

```

Explanation: • Performing BFS to find the minimum days required to heal all symptoms.

- Using a queue to explore states, where each state is a pair of the current symptom bitmask and the number of days taken.
- Using an `unordered_set` to track visited states to prevent reprocessing.