Question 3

- First we create a structure, Medicine that stores all the information relating to it, it's just to make my life easier.
- Going for the main function first, for each test case, we input the symptoms first and create an initialSymptoms using bitwise operations.
- We iterate through the symptoms and for each integer corresponding to 1, we make the i-th bit of initialSymptoms as 1 otherwise 0. This is to create a better handling of checking for symptoms.
- We then create an adjacency list or vector in this case and start assigning values similarly to the previous step. This creates the graph of the symptoms.
- The main function finally calls the solve functions and prints out the result.
- Coming to the solve() function, it is where all the computations happen. We know that the final state is when all diseases are cured, that is 0. So, we initialize it as such.
- We make a priority queue similar to Q2. We also create an unordered map which simply keeps track of the minimum distance to each node.
- Since we start from initialSymptoms, its value in the priority queue is 0 and the minDays map has its value as 0 too.
- Traversing the queue, if the currentState becomes 0, that is completely cured, we simply return the currentDays.
- If we haven't been cured yet, we iterate through the medicines. We take the bitwise and with the

complement of removes so that we cure those diseases and a bitwise or with the sideEffects to add those symptoms.