We repeat the following two steps until no vertex is deleted anymore. To check it, we record N before start. we check is the returned N is equal to previous N.

1.     We try to delete vertices that are adjacent with too few vertices. Specifically, we repeat following operation until the Boolean representing removal is false: Set a Boolean value to be false, meaning we haven’t removed vertex. For i from 1 to N, we count the number of times M(i, j) = 1 for j from 1 to N. If the number of times is less than 3, this vertex will not be included in sub-graph H. Change every M(i, j) to 0 and N to N - 1, meaning we remove vertex i. And we set the Boolean value to be true, meaning we did a removal.

2.     We try to delete vertices that are adjacent with too many vertices. Specifically, we repeat following operation until the Boolean representing removal is false: Set a Boolean value to be false, meaning we haven’t removed vertex. For i from 1 to N, we count the number of times M(i, j) = 0. If it is bigger than N-3, we move on to next i. If not, and if it is less than 3, this vertex will not be included in sub-graph H. Change every M(i, j) to 0 and N to N - 1, meaning we remove vertex i. And we set the Boolean value to be true, meaning we did a removal. Return N.

Now all vertices are eligible. If N is not 0, the remaining vertices and edges construct graph H. Otherwise, output a message “No such H can be found”.