Q No 4. [Local Search] **[5 Points]**

In this question we are going to pose the subset sum problem of described in first question as an optimization problem and then use Hill climbing strategy (i.e. a local search algorithm) to solve it.

Once again assume that for a set having n elements, a solution is coded using a bit string of length n with a bit being set to 1 if the element is part of the subset and 0 otherwise. Further, assume that the optimality of a solution is computed using $1/ (|S - \sum | + 1)$ where S is the required value of sum and $\sum$ is sum of the subset and $| x |$ represents absolute value of x.

A simple operator to generate a new solution from an existing solution can be defined as follows

NEW_SOLUTION(X) = FLIP A BIT IN THE SOLUTION X

This is equivalent to including an element in the subset or excluding an already chosen element from the subset. It is obvious that for a set of size **n** we can generate **n** new solutions from an existing solution.

Use the above operator for generating new solutions along with the hill climbing search strategy (also known as local search) to find a solution for the following subset sum problem.

Find a subset of the set {2, 3, 4, 8, 16} having sum **17**. Take the solution **00000** as the starting solution in your local search**.**

You must show all intermediate steps in the form of the following table. For each iteration show all intermediate solutions considered/generated and the solution selected at that iteration.

Iteration No	Intermediate Solutions	Selected Solution
1		
2		
. . .		