Steps.

1. Start with an empty set.
2. Add the next element from the list to the set.
3. If the subset is having sum M, then stop with that subset as solution.
4. If the subset is not feasible or if we have reached the end of the set, then backtrack through the subset until we find the most suitable value.
5. If the subset is feasible (Sum of subset <M), then go to step 2.
6. If we have visited all the elements without finding a suitable subset and if no backtracking is possible then stop without solution.

~~Pseudocode~~

Algorithm

```
Subset_sum (list, starting_index, target_sum)
{ if (target_sum == sum)
      { Subset_count ++;
        if (starting_index < list.length)
        { Subset_Sum (list, sum - list[starting_index -1],
                                  target_Sum); }
else
  { for (i= starting index; i< list.length, i++)
       { Subset_sum (list, sum+ list[i], target_sum);
       }
  } }
```