Combination Sum

Given an array of integers and a sum B, find all unique combinations in the array where the sum is equal to B. The same number may be chosen from the array any number of times to make B.

Note:

1. All numbers will be positive integers.

2. Elements in a combination (a1, a2, …, ak) must be in non-descending order. (ie, a1 ≤ a2 ≤ … ≤ ak).

3. The combinations themselves must be sorted in ascending order.

Example 1:

```
Input:
N = 4
arr[] = {7,2,6,5}
B = 16
Output:
(2 2 2 2 2 2 2 2)
(2 2 2 2 2 6)
(2 2 2 5 5)
(2 2 5 7)
(2 2 6 6)
(2 7 7)
(5 5 6)
```

Example 2:

```
Input:
N = 11
arr[] = {6,5,7,1,8,2,9,9,7,7,9}
B = 6
Output:
```

```
(1 1 1 1 1 1)
(1 1 1 1 2)
(1 1 2 2)
(1 5)
(2 2 2)
(6)
```

Your Task:
Your task is to complete the function **combinationSum()** which takes the array A and a sum B as inputs and returns a list of list denoting the required combinations in the order specified in the problem description. The printing is done by the driver's code. If no set can be formed with the given set, then "Empty" (without quotes) is printed.

Expected Time Complexity: $O(X^2 * 2^N)$, where X is average of summation B/arr_i for every number in the array.
Expected Auxiliary Space: $O(X * 2^N)$

Constraints:
1 <= N <= 30
1 <= A[i] <= 20
1 <= B <= 100