**Tania Chowdhury**

**Group\_num:01**

**Team members:**

*Brown,Jacob A  
Chowdhury,Tamzid  
Chowdhury,Tania  
Cortes,Victoria Danyel  
Deciutiis II,James Hugo*

**Question 8:**

**Approach:**

Some elements can be repeated in the given array. We make sure to iterate over the number of occurrences of those elements to avoid repeated combinations. Once we have done that, things are fairly straightforward. We Call a [recursive](https://www.geeksforgeeks.org/recursion/) function **uniqueCombination()** with the remaining sum and make the indices to move forward. When the sum of a combination matches the given sum,we print all the elements which were selected to get this sum.

**Template class:**

We make a template class **SumCombination** no data members and a public function **findAllCombination(vector<T> vec, T sum)** that print all the combinations that add up to the given sum**.** If this function can not find any combination it prints **“NO COMBINATION FOUND**". The **findAllCombination** make use of a recursive function **uniqueCombination(int, T, T, vector<T>&, vector<T>&, bool&)** to achieve the task. This function print all combinations and sets **find** true else if can not find any combination that adds upto the given sum that **find** remains **false**