**3. Code Question 2**

Amazon Shopping has ***n* items** in inventory.

Each item has a rating that may be negative.

The team wants to work on an algorithm that will suggest combinations of these items that customers might buy, or, *combos* for short. A combo is defined as a subset of the given *n* items.

The total popularity of a combo is the sum of the popularities of the individual items in the combo. Design an algorithm that can find the *k* combos with the highest popularities.

Two combos are considered different if they have a different subset of items.

Return an array of *k* integers where the *ith*denotes the popularity of *ith* best combo.

Combos should be returned arranged best to worst.

**Note:** You can have an empty subset as a combo as well. The popularity for such a subset is 0.

**Example**

*n* = 3

*popularity* = [3, 5, -2]

*k* = 3

All possible popularities of combos are 0, 3, 5, -2, 8, 3, 1, 6. The best three combos have popularities 8, 6, and 5. The answer is [8, 6, 5].

**Function Description**

Complete the function *bestCombos* in the editor below.

*bestCombos* has the following parameters:

*int popularity[n]:*  *popularity[i]* denotes the popularity of *ith* item (0 ≤ *i* < *n*).

*int k:* the number of best combos to return

**Returns**

*int[k]:* the popularities of the best *k* combos, in decreasing order

**Constraints**

* 1 ≤ n ≤ 105
* -109 ≤ *popularity[i*] ≤ 109
* 1 ≤ *k* ≤ *min*(2000, 2n)