**Given an array and target 'x', Find out the pair from array whose sum forms the number 'x' when both the numbers of the pair are added) :**

Given an array *arr1 and number 'x',* To find if a pair exists in *arr1* which can be added to make number 'x'.

**Example-1:**

**Input:**

arr1 = {0, -1, 2, -3, 1}

x = -2

**Output:** true

(-3, 1) or (1,-3) are the pairs which exist in arr1 which makes sum equal to x (-3+1=-2 and 1-3=-2)

**Explanation**: Calculate the sum of all possible pairs and return (true/false) if the value of any pair is equal/not equal to 'x'.

**Example-2:**

**Input:**

arr1 = {1, -2, 1, 0, 5}

x = 0

**Output:** false

**Explanation**: No sum of pairs in array has sum as 0 , therefore , false will be the answer

**private static boolean twoSum(int [] nums , int target){**

**int n = nums.length();**

**//hash Map to store frequencies**

**Set<Integer > set = new HashSet<>();**

**for(int i = 0 ; i < nums.length ; i++){**

**// Find the rem value required to make the target**

**int required = target - nums[i];**

**// If it is present then return true**

**if(set.contains(required))**

**return true;**

**// Add the value to the set**

**set.add(nums[i]);**

**}**

**return false;**

**}**