## Two sum

Given an array of integers nums and an integer target, return indices of 2 numbers such that they add up to target. You may assume that each input would have exactly one solution, and you may not use some element twice. You can return the answer in any order.

Ex:
Input: nums = [3, 2, 4], target = 6
o/p: [1, 2]

**constraints**

- $2 <= nums.length <= 10^4$
- $-10^9 <= num[i] <= 10^9$
- $-10^9 <= target <= 10^9$

**Logic:**

- intialize a variable sum to 0
- get the length of array and store it in a variable n
- use two for loops to check every possible pair of numbers in the array
  - The outer loop iterates through first element (index i=0 to i=n-1)
  - The inner loop iterates through second element (index j=i+1 to n-1).
- for each pair of elements calculate sum ; sum = nums[i] + nums[j]
- check if sum is equal to target
- if sum == target, break the loop & return the indices
- if no pair sum matches target. return empty array.

```
class solution {
    public int[] twosum(int[] nums, int target){
        int sum=0;
        int n=nums.length;
        for(int i=0;i<n;i++){
            for(int j=i+1;j<n;j++){
                sum=nums[i]+nums[j];
                if(sum==target){
                    return new int[]{i,j};
                }
            }
        }
        return new int[]{};
    }
```