

Given an array of integers `nums` and an integer `target`, return indices of the two numbers such that they add up to `target`.

You may assume that each input would have exactly one solution, and you may not use the same element twice.

You can return the answer in any order.

**Example 1:**

**Input:** `nums = [2,7,11,15]`, `target = 9`

**Output:** `[0,1]`

**Explanation:** Because `nums[0] + nums[1] == 9`, we return `[0, 1]`.

## Brute Force Approach:

```
import java.io.*;
import java.util.*;

class Main{

    public static void main(String args[]){

        int arr[]={1,2,3,5};
        int target=4;

        Main m = new Main();

        System.out.println(Arrays.toString(TwoSum(arr,target)));
    }

    public static int[] TwoSum(int arr[],int target)
    {
        for(int i=0;i<arr.length;i++)
        {
            for(int j=i+1;j<arr.length;j++)
            {
                if(arr[j]==target-arr[i])
                {
                    return new int[]{i,j};
                }
            }
        }

        return null;
    }
}
```