Given an array of integers,
Return the **indices** of
two numbers such that they
add up to a specific target

Reasonable Assumptions:

** You can assume that each
** input has exactly 1 solution

** You can't use the exact
** same element

```python
def TwoSum(nums,target):
    temp = {}

    for index, value in enumerate(num):
        if target-value in temp:
            return [temp[target-value],index]
        else:
            temp[value]=index
```

Breakdown:

example input = [2,7,11,15]   Target = 26

temp = { }   // initially empty

for index,value in enumerate(nums):

*remember what enumerate does!
essentially this is what we're doing

index	0	1	2	3
value	2	7	11	15

The first iteration   index = 0
                       value = 2

if target - num in temp:
  └> if (26 - 2) in { } # False
else:
  a[num] = i
  └> a[2] = 0 , meaning a = {2:0}
     2 is the key — 0 is the value

* Remember in
Python dictionaries
are
Key : value

We want 2 to be the key
B/C the problem is asking us to
the **INDICIES**

1	2	3
7	11	15

return

2nd iteration

if target - num in temp:

$26 - 7$ in $\{2:0\}$ // False

else

temp $[7] = 1$

$\hookrightarrow$ $\{2:0, 7:1\}$

2	3
11	15

3rd iteration

if target - num in temp:

$26 - 11$ in $\{2:0, 7:1\}$ // _False_!

else

temp $[11] = 2$

$\hookrightarrow$ $\{2:0, 7:1, 11:2\}$

remainder of the input array → $\dfrac{3}{15}$

4th iteration:

if target - num in temp:

$26 - 15$ in $\{2:0, 7:1, 11:2\}$ // _TRUE_

11 is in the dictionary Temp

return [Temp $[26-15]$, index)

Temp $[11]$, 3

return $(2, 3)$ ——◇ Solved!