

# Remove

Problem ID: a07p08remove

In order to move a disc, we need to remove it from one location and insert it at another. For example, imagine that all the discs are on the first pillar and we want to move the smallest disc to the third pillar. That means that we would like to change our state representation from `321|||` to `32||1|` by moving the 1 within the string. The code to achieve that could look something like this:

We start in the state: `321|||`

```
• where_to_remove = find_index_of_nth_occurrence(
    sequence=state,
    element_to_find="|",
    occurrence=1
) - 1

• state, disc_to_move = remove_at(
    sequence=state,
    index_to_remove=where_to_remove
)

• where_to_insert = find_index_of_nth_occurrence(
    sequence=state,
    element_to_find="|",
    occurrence=3
)

• state = insert_at(
    sequence=state,
    index=where_to_insert,
    element=disc_to_move
)
```

Our final state is now `32||1|`

In this exercise, you will implement the `remove_at` function. It is a bit special in that it returns two values; the updated sequence and the element that was removed.

**Note that we are testing your code differently in this task, please only submit your function definitions, without any code outside the functions!** The main python file, which handles input and output, is already provided. You can download and place the main file in the same directory as your python file. You can then run the main python file we provide to try out the samples.

## Input

The function receives two parameters, a sequence  $s$  and an integer  $i$ . In the tests,  $s$  will be a string with  $1 \leq |s| \leq 20$ , and  $i$  will be an index pointing to a position in  $s$ , so  $0 \leq i < |s|$ .

It is good if your function also works for other types of sequences and elements, or for input outside these specifications, but that is not part of the requirements.

In the samples below, the first line of the input contains the string  $s$  and the second line contains the index  $i$ .

## Output

The function should return a pair  $(s', e)$  of two values. The first value  $s'$  should be a sequence identical to the input sequence  $s$  except with the element at index  $i$  removed, so  $|s'| = |s| - 1$ , and the second value  $e$  should be the element that used to be at index  $i$  in the input sequence.

So given a string  $s$  as input (as in the tests),  $s'$  should also be a string, and  $e$  should be a character from  $s$ , which in Python is just a string of length 1.

Note that to return a pair of values, you can just separate them with a `,` in a single `return` statement, as in:

```
return updated_sequence, removed_element
```

In the samples below, the first and only line of the output contains the pair  $(s', e)$ .

**Sample Input 1**

321|||  
2

**Sample Output 1**

('32|||', '1')

**Sample Input 2**

3|21||  
0

**Sample Output 2**

('|21||', '3')

**Sample Input 3**

3|21||  
5

**Sample Output 3**

('3|21|', '|')