

## Challenge 3 - YATM Microservice

[« Prev](#) [Next »](#)

At Tuenti, we have a degree of freedom to choose the language of different microservices. Each microservice is based on what the developer knows best and what they think is the most suitable platform for the thing they are solving. We recently discovered a microservice running on an old platform, and we want to create a docker image in order to replace it with newer instances. We were able to extract the code from the machines and we also dumped the contents, which originally looked like this:



We have also saved what the original machine was returning for those tapes, which could be useful for debugging. So we need you to help us write an interpreter for that code.

### Input

The input will contain the machine code and the contents of the tapes in another well-known format.

### Output

The final state for each tape, each on a different line and starting with **Tape #N:**

## Sample Input

```
---
code:
  start:
    '#':
      move: right
      state: replace
  replace:
    '0':
      write: '1'
      move: right
    '1':
      state: end
    ' ':
      write: '#'
      state: end
tapes:
  1: '#001'
  2: '#0001'
  3: '#0'
  4: '#1'
  ...
```

## Sample Output

```
Tape #1: #111
Tape #2: #1111
Tape #3: #1#
Tape #4: #1
```

## Test your code

---

You can test your program against both the input provided in the test phase and the input provided in the submit phase. A nice output will tell you if your program got the right solution or not. You can try as many times as you want to. Be careful with extra whitespaces, the output should be exactly as described.

### Test your program against the input provided in the test phase

[Download test input](#)

Program output:

 Ningún archivo seleccionado

## Test your program against the input provided in the submit phase

[Download input](#)

Program output:

 Ningún archivo seleccionado

During the submit phase, in some problems, we might give your program harder inputs. As with the test token, a nice output will tell you if your program got the right solution or not. You can try as many times as you need.

In the actual contest you first need to solve the test phase before submitting the code, you must provide the source code used to solve the challenge and you can only submit once (once your solution is submitted you won't be able to amend it to fix issues or make it faster).

If you have any doubts, please check the [info section](#).

[« Prev](#)   [Next »](#)

---

Tweet about this! [#TuentiChallenge6](#)

[Follow @TuentiEng](#)