



input: '000111'  
blank: ''  
start state: right  
table:  
right:  
0: R  
1: {R: accept}  
': {L: left}  
left:  
0: L  
1: {L: reject}  
': {R: right}  
accept:  
  
reject:

♥ Mi piace

Modifica la Turing Machine in modo che decida il linguaggio delle sequenze di 0 che sono potenze di 3:  $\{0^{3^n} \mid n \geq 0\}$

input: '000000'  
blank: ''  
start state: start  
table:  
start:  
0: {write: '', R: one}  
': {R: reject}  
one:  
x: R  
0: {write: x, R: two}  
': {R: accept}  
two:  
x: R  
0: {write: x, R: three}  
': {R: reject}  
three:  
x: R  
0: {R: jump}  
': {L: back}  
jump:  
x: R  
0: {write: x, R: two}  
': {R: reject}  
back:  
[0, x]: L  
': {R: one}  
accept:  
reject:

♥ Mi piace

Modifica la Turing Machine in modo che decida il linguaggio delle coppie di parole dove la seconda stringa è il complemento a 1 della prima (0 e 1 scambiati), per esempio '1101#0010'

```
input: '1101#0010'
blank: ''
start state: start
table:
  start:
    [0,1]: R
    '#': {L: rollback}
    ' ': {R: reject}
  rollback:
    [0,1]: L
    ' ': {R: check}
  check:
    0: {write: 'x', R: zero}
    1: {write: 'x', R: one}
    '#': {R: continue}
  zero:
    [0,1]: R
    '#': {R: is_one}
  one:
    [0,1]: R
    '#': {R: is_zero}
  is_zero:
    'x': R
    0: {write: 'x', L: back1}
  is_one:
    'x': R
    1: {write: 'x', L: back1}
  back1:
    'x': L
    '#': {L: back2}
  back2:
    [0,1]: L
    'x': {R: check}
  continue:
    'x': R
    ' ': {R: accept}
    [0,1]: {R: reject}
  accept:
  reject:
```

♥ Mi piace