

1 Advent of Code 2025 in Typst 😊

2 Contents

3	Advent of Code 2025 in Typst 😊	1
4	day 1	1
5	python	1
6	typst	3

7 day 1

8 python

```
9 >>> from main import xs
10
11 >>> xs
12 '\nL68\nL30\nR48\nL5\nR60\nL55\nL1\nL99\nR14\nL82\n'
13
14 >>> xs.split()
15 ['L68', 'L30', 'R48', 'L5', 'R60', 'L55', 'L1', 'L99', 'R14', 'L82']
16
17 >>> xs.replace("L", "+")
18 '\n+68\n+30\nR48\n+5\nR60\n+55\n+1\n+99\nR14\n+82\n'
19
20 >>> xs.replace("L", "+").replace("R", "-")
21 '\n+68\n+30\n-48\n+5\n-60\n+55\n+1\n+99\n-14\n+82\n'
22
23 >>> xs.replace("L", "+").replace("R", "-").split()
24 ['+68', '+30', '-48', '+5', '-60', '+55', '+1', '+99', '-14', '+82']
25
26 >>> [int(e) for e in xs.replace("L", "+").replace("R", "-").split()]
27 [68, 30, -48, 5, -60, 55, 1, 99, -14, 82]
28
29 >>> list(map(int, xs.replace("L", "+").replace("R", "-").split()))
30 [68, 30, -48, 5, -60, 55, 1, 99, -14, 82]
31
32 >>> (50 + 68) % 100
33
34 18
35
36 >>> (50 - 68) % 100
37
38 82
39
40 >>>
41 >>> temp = 50
42 >>> for r in map(int, xs.replace("L", "-").replace("R", "+").split()):
43 ...     temp
44 KeyboardInterrupt
45
46 >>>
47 KeyboardInterrupt
48
49 >>> temp = 50
50 >>> count = 0
51
52 >>>
```

```
52 >>>
53 >>>
54 >>> for r in map(int, xs.replace("L", "-").replace("R", "+").split()):
55 ...     print(temp)
56 ...     temp = (temp + r) % 100
57 ...     print(temp)
58 ...     if temp == 0: count += 1
59 ...     print(count)
60 ...
61 50
62 82
63 0
64 82
65 52
66 0
67 52
68 0
69 1
70 0
71 95
72 1
73 95
74 55
75 1
76 55
77 0
78 2
79 0
80 99
81 2
82 99
83 0
84 3
85 0
86 14
87 3
88 14
89 32
90 3
91 >>>
92 >>>
93 >>> (lambda xs: (
94 ...     lambda pos, c:
95 ...        ([(pos := (pos + r) % 100,
96 ...             c := c + (pos == 0))
97 ...          for r in map(int, xs.replace("L", "-").replace("R", "+").split())]),
98 ...         c)[1]
99 ... )(50, 0))(xs)
100 3
101 >>>
102 >>>
```

Typst → Python

```
103 typst
104
105 #let count-zeros(xs) = {
106   let steps = xs.replace("L", "-").replace("R", "+").split().map(int)
107   let result = steps.fold((50, 0), (acc, r) => {
108     let pos = calc.rem((acc.at(0) + r), 100)
109     let cnt = acc.at(1) + int(pos == 0)
110     (pos, cnt)
111   })
112   result.at(1)
113 }
114
115 #count-zeros("L68 L30 R48 L5 R60 L55 L1 L99 R14 L82")
116 3
```

Typst ↗ Python