

Advent of Code [About] [Events] [Shop] [Settings] [Log Out] mehaase 42*
 int y=2018; [Calendar] [AoC++] [Sponsors] [Leaderboard] [Stats]

--- Day 14: Chocolate Charts ---

You finally have a chance to look at all of the produce moving around. Chocolate, cinnamon, mint, chili peppers, nutmeg, vanilla... the Elves must be growing these plants to make hot chocolate! As you realize this, you hear a conversation in the distance. When you go to investigate, you discover two Elves in what appears to be a makeshift underground kitchen/laboratory.

The Elves are trying to come up with the ultimate hot chocolate recipe; they're even maintaining a scoreboard which tracks the quality score (0-9) of each recipe.

Only two recipes are on the board: the first recipe got a score of 3, the second, 7. Each of the two Elves has a current recipe: the first Elf starts with the first recipe, and the second Elf starts with the second recipe.

To create new recipes, the two Elves combine their current recipes. This creates new recipes from the digits of the sum of the current recipes' scores. With the current recipes' scores of 3 and 7, their sum is 10, and so two new recipes would be created: the first with score 1 and the second with score 0. If the current recipes' scores were 2 and 3, the sum, 5, would only create one recipe (with a score of 5) with its single digit.

The new recipes are added to the end of the scoreboard in the order they are created. So, after the first round, the scoreboard is 3, 7, 1, 0.

After all new recipes are added to the scoreboard, each Elf picks a new current recipe. To do this, the Elf steps forward through the scoreboard a number of recipes equal to 1 plus the score of their current recipe. So, after the first round, the first Elf moves forward $1 + 3 = 4$ times, while the second Elf moves forward $1 + 7 = 8$ times. If they run out of recipes, they loop back around to the beginning. After the first round, both Elves happen to loop around until they land on the same recipe that they had in the beginning; in general, they will move to different recipes.

Drawing the first Elf as parentheses and the second Elf as square brackets, they continue this process:

```
(3)[7]
(3)[7] 1 0
3 7 1 [0](1) 0
3 7 1 0 [1] 0 (1)
(3) 7 1 0 1 0 [1] 2
3 7 1 0 (1) 0 1 2 [4]
3 7 1 [0] 1 0 (1) 2 4 5
3 7 1 0 [1] 0 1 2 (4) 5 1
3 (7) 1 0 1 0 [1] 2 4 5 1 5
3 7 1 0 1 0 1 2 [4](5) 1 5 8
3 (7) 1 0 1 0 1 2 4 5 1 5 8 [9]
3 7 1 0 1 0 1 [2] 4 (5) 1 5 8 9 1 6
3 7 1 0 1 0 1 2 4 5 [1] 5 8 9 1 (6) 7
3 7 1 0 (1) 0 1 2 4 5 1 5 [8] 9 1 6 7 7
3 7 [1] 0 1 0 (1) 2 4 5 1 5 8 9 1 6 7 7 9
3 7 1 0 [1] 0 1 2 (4) 5 1 5 8 9 1 6 7 7 9 2
```

The Elves think their skill will improve after making a few recipes (your puzzle input). However, that could take ages; you can speed this up considerably by identifying the scores of the ten recipes after that. For example:

- If the Elves think their skill will improve after making 9 recipes,

Our sponsors help
make Advent of
Code possible:

SmartyStreets -
Global address
validation made
by developers,
for developers

- the scores of the ten recipes after the first nine on the scoreboard would be `5158916779` (highlighted in the last line of the diagram).
- After `5` recipes, the scores of the next ten would be `0124515891`.
 - After `18` recipes, the scores of the next ten would be `9251071085`.
 - After `2018` recipes, the scores of the next ten would be `5941429882`.

What are the scores of the ten recipes immediately after the number of recipes in your puzzle input?

Your puzzle answer was `7861362411`.

--- Part Two ---

As it turns out, you got the Elves' plan backwards. They actually want to know how many recipes appear on the scoreboard to the left of the first recipes whose scores are the digits from your puzzle input.

- `51589` first appears after `9` recipes.
- `01245` first appears after `5` recipes.
- `92510` first appears after `18` recipes.
- `59414` first appears after `2018` recipes.

How many recipes appear on the scoreboard to the left of the score sequence in your puzzle input?

Your puzzle answer was `20203532`.

Both parts of this puzzle are complete! They provide two gold stars: **

At this point, you should [return to your advent calendar](#) and try another puzzle.

Your puzzle input was `919901`.

You can also [\[Share\]](#) this puzzle.