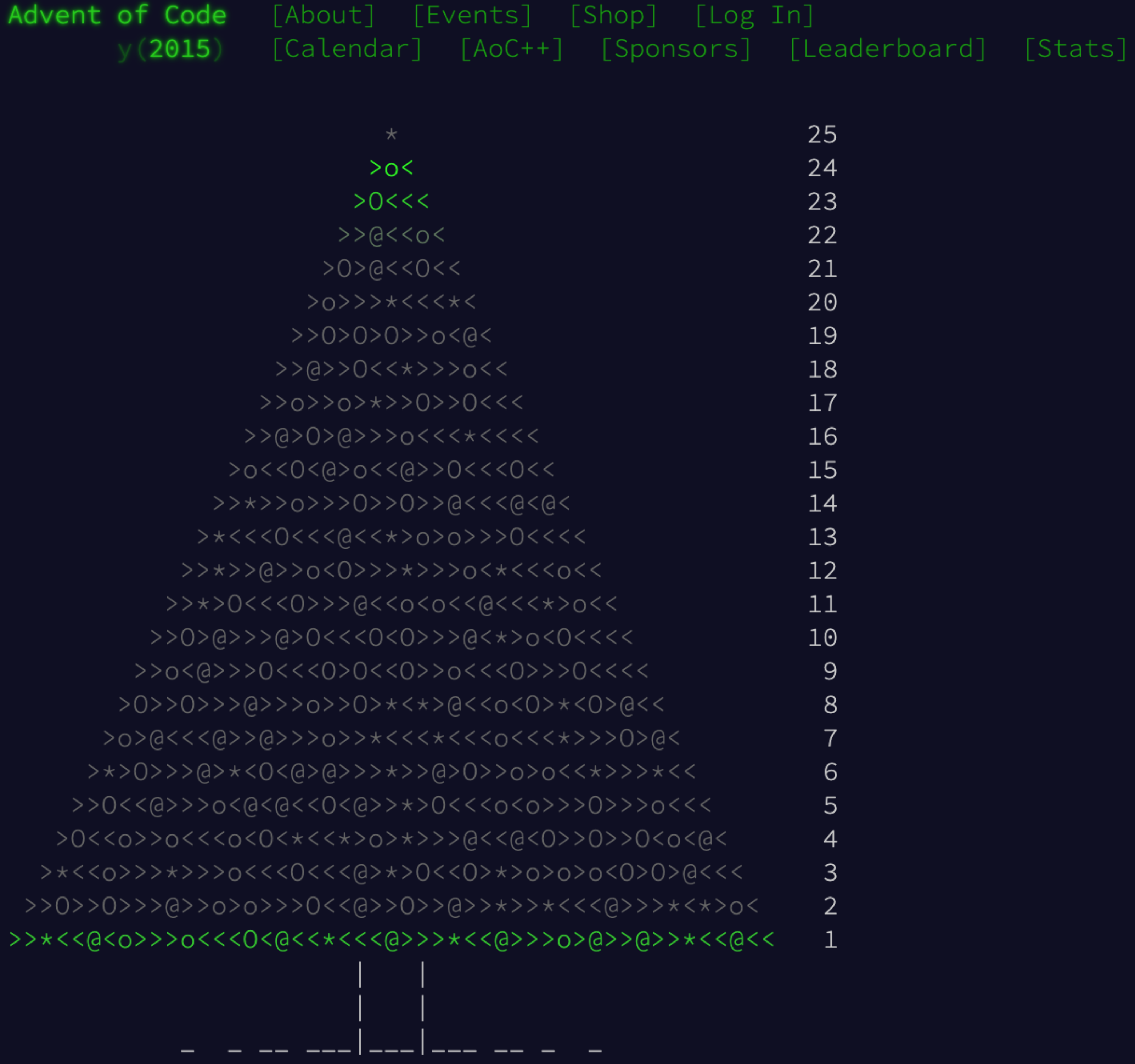
I did 'Advent of Code' and here is what I learned

SoCalRUG 2022

Emil Hvitfeldt



```
                        *                          25
                       >o<                         24
                      >O<<<                        23
                     >>@<<o<                       22
                    >O>@<<O<<                      21
                   >o>>>*<<<*<                     20
                  >>O>O>O>>o<@<                    19
                 >>@>>O<<*>>>o<<                   18
                >>o>>o>*>>O>>O<<<                  17
               >>@>O>@>>>o<<<*<<<<                 16
              >o<<O<@>o<<@>>O<<<O<<                15
             >>*>>o>>>O>>O>>@<<<@<@<               14
            >*<<<O<<<@<<*>o>o>>>O<<<<              13
           >>*>>@>>o<O>>>*>>>o<*<<<o<<             12
          >>*>O<<<O>>>@<<o<o<<@<<<*>o<<            11
         >>O>@>>>@>O<<<O<O>>>@<*>o<O<<<<           10
        >>o<@>>>O<<<O>O<<O>>o<<<O>>>O<<<<           9
       >O>>O>>>@>>>o>>O>*<*>@<<o<O>*<O>@<<          8
      >o>@<<<@>>@>>>o>>*<<<*<<<o<<<*>>>O>@<         7
     >*>O>>>@>*<O<@>@>>>*>>@>O>>o>o<<*>>>*<<        6
    >>O<<@>>>o<@<@<<O<@>>*>O<<<o<o>>>O>>>o<<<       5
   >O<<o>>o<<<o<O<*<<*>o>*>>>@<<@<O>>O>>O<o<@<      4
  >*<<o>>>*>>>o<<<O<<<@>*>O<<O>*>o>o>o<O>O>@<<<     3
 >>O>>O>>>@>>o>o>>>O<<@>>O>>@>>*>>*<<<@>>>*<*>o<    2
>>*<<@<o>>>o<<<O<@<<*<<<@>>>*<<@>>>o>@>>@>>*<<@<<   1
                      |   |
                      |   |
           _  _ __ ___|___|___ __ _  _
```

--- Day 2: Password Philosophy ---

Your flight departs in a few days from the coastal airport; the easiest way down to the coast from here is via toboggan.

The shopkeeper at the North Pole Toboggan Rental Shop is having a bad day. "Something's wrong with our computers; we can't log in!" You ask if you can take a look.

Their password database seems to be a little corrupted: some of the passwords wouldn't have been allowed by the Official Toboggan Corporate Policy that was in effect when they were chosen.

To try to debug the problem, they have created a list (your puzzle input) of **passwords** (according to the corrupted database) and **the corporate policy when that password was set**.

For example, suppose you have the following list:

```
1-3 a: abcde
1-3 b: cdefg
2-9 c: ccccccccc
```

Each line gives the password policy and then the password. The password policy indicates the lowest and highest number of times a given letter must appear for the password to be valid. For example, `1-3 a` means that the password must contain `a` at least `1` time and at most `3` times.

In the above example, `2` passwords are valid. The middle password, `cdefg`, is not; it contains no instances of `b`, but needs at least `1`. The first and third passwords are valid: they contain one `a` or nine `c`, both within the limits of their respective policies.

How many passwords are valid according to their policies?

To play, please identify yourself via one of these services:

[GitHub] [Google] [Twitter] [Reddit] - [How Does Auth Work?]

--- Day 2: Password Philosophy ---

Your flight departs in a few days from the coastal airport; the easiest way down to the coast from here is via toboggan.

The shopkeeper at the North Pole Toboggan Rental Shop is having a bad day. "Something's wrong with our computers; we can't log in!" You ask if you can take a look.

Their password database seems to be a little corrupted: some of the passwords wouldn't have been allowed by the Official Toboggan Corporate Policy that was in effect when they were chosen.

To try to debug the problem, they have created a list (your puzzle input) of **passwords** (according to the corrupted database) and **the corporate policy when that password was set**.

For example, suppose you have the following list:

```
1-3 a: abcde
1-3 b: cdefg
2-9 c: ccccccccc
```

Each line gives the password policy and then the password. The password policy indicates the lowest and highest number of times a given letter must appear for the password to be valid. For example, `1-3 a` means that the password must contain `a` at least `1` time and at most `3` times.

In the above example, `2` passwords are valid. The middle password, `cdefg`, is not; it contains no instances of `b`, but needs at least `1`. The first and third passwords are valid: they contain one `a` or nine `c`, both within the limits of their respective policies.

How many passwords are valid according to their policies?

To play, please identify yourself via one of these services:

--- Day 2: Password Philosophy ---

Your flight departs in a few days from the coastal airport; the easiest way down to the coast from here is via toboggan.

The shopkeeper at the North Pole Toboggan Rental Shop is having a bad day. "Something's wrong with our computers; we can't log in!" You ask if you can take a look.

Their password database seems to be a little corrupted: some of the passwords wouldn't have been allowed by the Official Toboggan Corporate Policy that was in effect when they were chosen.

To try to debug the problem, they have created a list (your puzzle input) of **passwords** (according to the corrupted database) and **the corporate policy when that password was set**.

For example, suppose you have the following list:

```
1-3 a: abcde
1-3 b: cdefg
2-9 c: ccccccccc
```

Each line gives the password policy and then the password. The password policy indicates the lowest and highest number of times a given letter must appear for the password to be valid. For example, `1-3 a` means that the password must contain `a` at least `1` time and at most `3` times.

In the above example, `2` passwords are valid. The middle password, `cdefg`, is not; it contains no instances of `b`, but needs at least `1`. The first and third passwords are valid: they contain one `a` or nine `c`, both within the limits of their respective policies.

How many passwords are valid according to their policies?

To play, please identify yourself via one of these services:

[GitHub] [Google] [Twitter] [Reddit] - [How Does Auth Work?]

--- Day 2: Password Philosophy ---

Your flight departs in a few days from the coastal airport; the easiest way down to the coast from here is via toboggan.

The shopkeeper at the North Pole Toboggan Rental Shop is having a bad day. "Something's wrong with our computers; we can't log in!" You ask if you can take a look.

Their password database seems to be a little corrupted: some of the passwords wouldn't have been allowed by the Official Toboggan Corporate Policy that was in effect when they were chosen.

To try to debug the problem, they have created a list (your puzzle input) of **passwords** (according to the corrupted database) and **the corporate policy when that password was set**.

For example, suppose you have the following list:

```
1-3 a: abcde
1-3 b: cdefg
2-9 c: ccccccccc
```

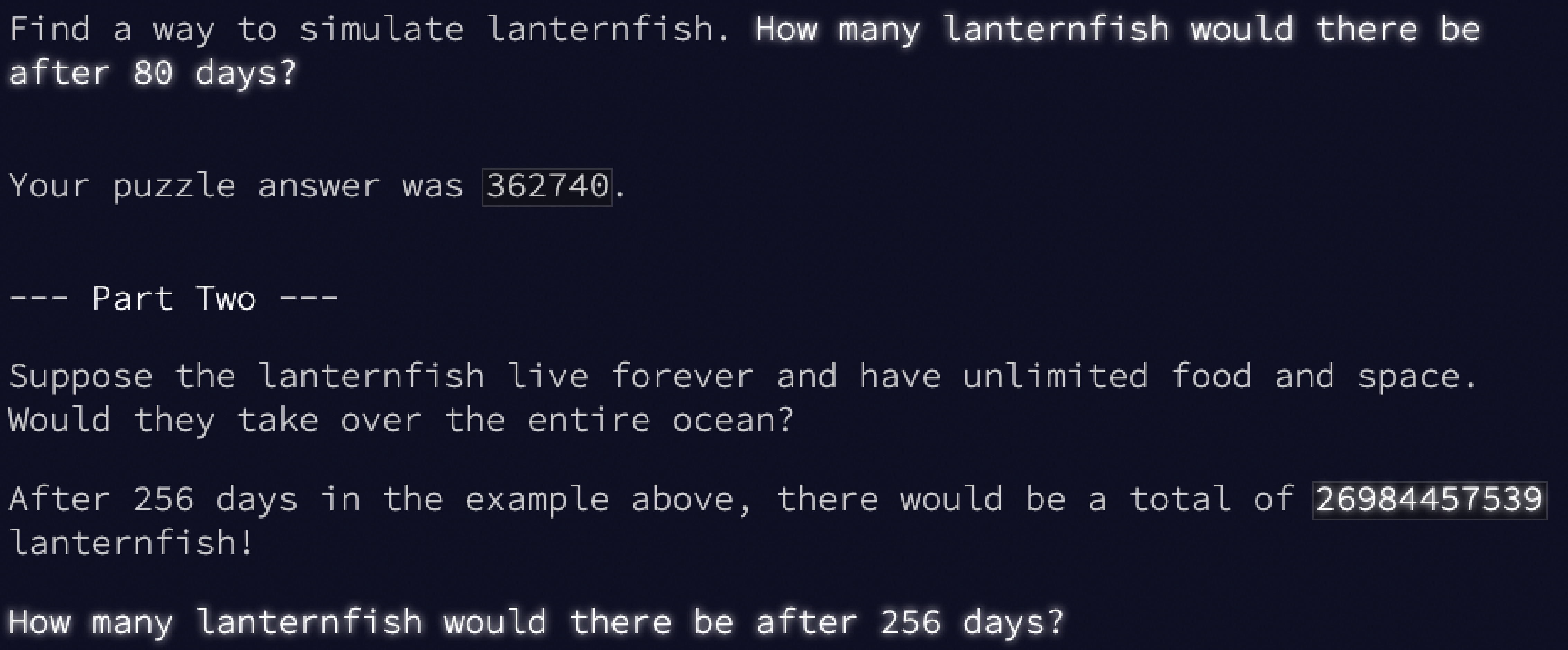
Each line gives the password policy and then the password. The password policy indicates the lowest and highest number of times a given letter must appear for the password to be valid. For example, `1-3 a` means that the password must contain `a` at least `1` time and at most `3` times.

In the above example, **`2`** passwords are valid. The middle password, `cdefg`, is not; it contains no instances of `b`, but needs at least `1`. The first and third passwords are valid: they contain one `a` or nine `c`, both within the limits of their respective policies.

How many passwords are valid according to their policies?

To play, please identify yourself via one of these services:

[GitHub] [Google] [Twitter] [Reddit] - [How Does Auth Work?]



Find a way to simulate lanternfish. **How many lanternfish would there be after 80 days?**

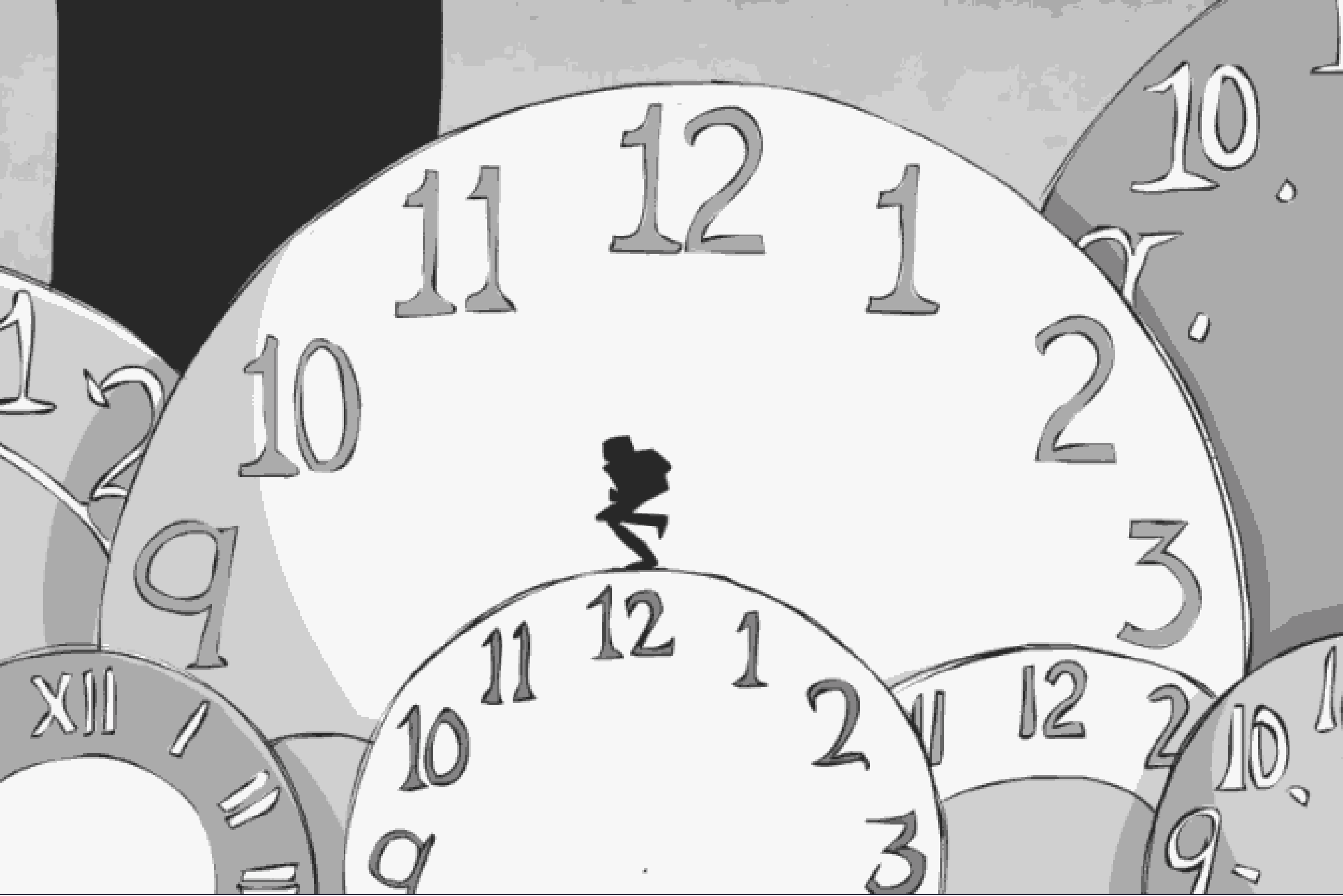
Your puzzle answer was `362740`.

--- Part Two ---

Suppose the lanternfish live forever and have unlimited food and space. Would they take over the entire ocean?

After 256 days in the example above, there would be a total of **`26984457539`** lanternfish!

How many lanternfish would there be after 256 days?



2020 December 1st

Find the two entries that sum to 2020

My Solution

```r
input <- as.numeric(readLines("01-input"))

sum2 <- function(input, target) {
  for (i in input) for (j in input) if (i + j == target) return(i * j)
}

sum2(input, 2020)
```

502 Bad Gateway 🔥🔥🔥

Possible Approaches

- loops
- expand.grid
- sample
- Rockstar

```
The goal is to extinguish my desparation
Knock the goal down

The enemy are accountants
Knock the enemy down

The system is the enemy
The machine is the enemy

Listen to the money
Until the money is gone
Cast the money into the fire
Let the rage at the enemy be the fire
Build the enemy up
Listen to the money

While the enemy is higher than the system
Let chaos be the rage at the system
Let destruction be the rage at the machine
If the goal is chaos with destruction
Break it down

Build the machine up
If the enemy is higher than the machine
Take it to the top

Build the system up
The machine is the system

Shout chaos of destruction
```

https://old.reddit.com/r/adventofcode/comments/k4e4lm/2020 day 1 solutions/ge8uwqi/

Types of problems

- recursive
- Grid based
- parsable

Parsable Solutions

```
forward 2
down 4
down 3
up 4
down 1
down 8
up 9
forward 1
down 9
forward 6
down 7
forward 1
down 1
up 2
forward 8
down 3
down 9
down 3
```

Parsable Solutions

```
forward(2)
down(4)
down(3)
up(4)
down(1)
down(8)
up(9)
forward(1)
down(9)
forward(6)
down(7)
forward(1)
down(1)
up(2)
forward(8)
down(3)
down(9)
down(3)
```

- modify the input
  - define `forward()`, `down()` and `up()`
  - run `eval(parse(text = input))`
  - …
  - profit

```
memory exhausted
(limit reached)
```

Lanternfish

```
Initial state: 3,4,3,1,2
```

Lanternfish

```
Initial state: 3,4,3,1,2
After  1 day:  2,3,2,0,1
```

Lanternfish

```
Initial state: 3,4,3,1,2
After  1 day:  2,3,2,0,1
After  2 days: 1,2,1,6,0,8
```

Lanternfish

```
Initial state: 3,4,3,1,2
After  1 day:  2,3,2,0,1
After  2 days: 1,2,1,6,0,8
After  3 days: 0,1,0,5,6,7,8
After  4 days: 6,0,6,4,5,6,7,8,8
After  5 days: 5,6,5,3,4,5,6,7,7,8
After  6 days: 4,5,4,2,3,4,5,6,6,7
After  7 days: 3,4,3,1,2,3,4,5,5,6
After  8 days: 2,3,2,0,1,2,3,4,4,5
After  9 days: 1,2,1,6,0,1,2,3,3,4,8
After 10 days: 0,1,0,5,6,0,1,2,2,3,7,8
```

Lanternfish

Part 1

How many lanternfish would there be after 80 days?

Lanternfish

Part 1

How many lanternfish would there be after 80 days?

Part 2

How many lanternfish would there be after 256 days?

Lanternfish

Part 1

How many lanternfish would there be after 80 days? ~1.5 MB

Part 2

How many lanternfish would there be after 256 days?

Lanternfish

Part 1

How many lanternfish would there be after 80 days? **~1.5 MB**

Part 2

How many lanternfish would there be after 256 days? **~6.58 TB**

General workflow

- reading in data
- doing calculations
- unit tests
- Solution

- {readr}
- `data.table::fread()`
- {jsonlite}
- `read_parquet()`

General workflow

- reading in data
- **doing calculations**
- unit tests
- Solution

- vectors
- matrices
- data.frames
- lists

General workflow

- reading in data
- doing calculations
- **unit tests**
- Solution

General workflow

- reading in data
- doing calculations
- unit tests
- Solution

You might learn something

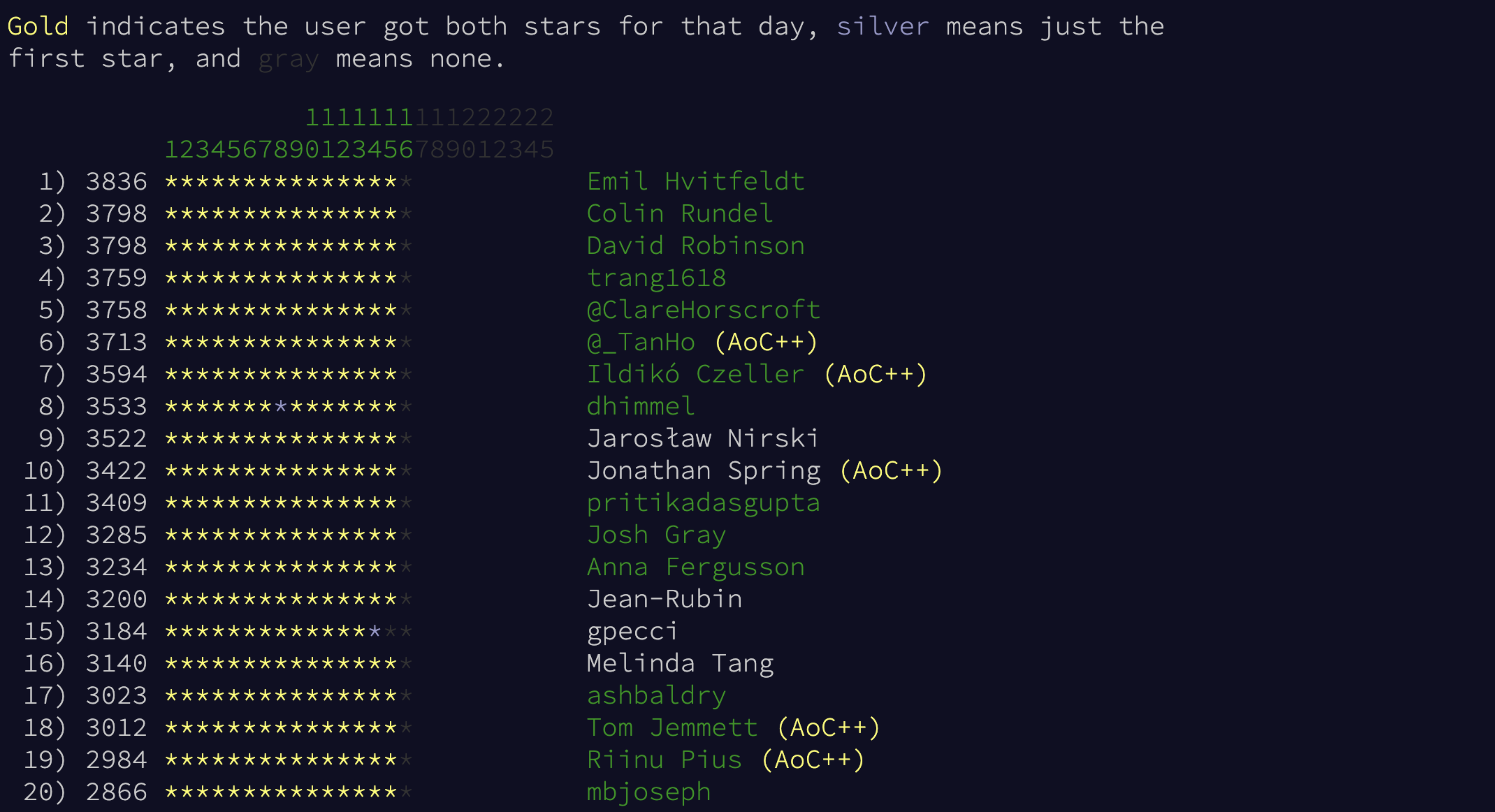
Ways you can play this game

- You can go for speed
- shortest solution
- no dependencies
- try another language

SLOW
FAST
FASTEST
- EXCUSE ME.

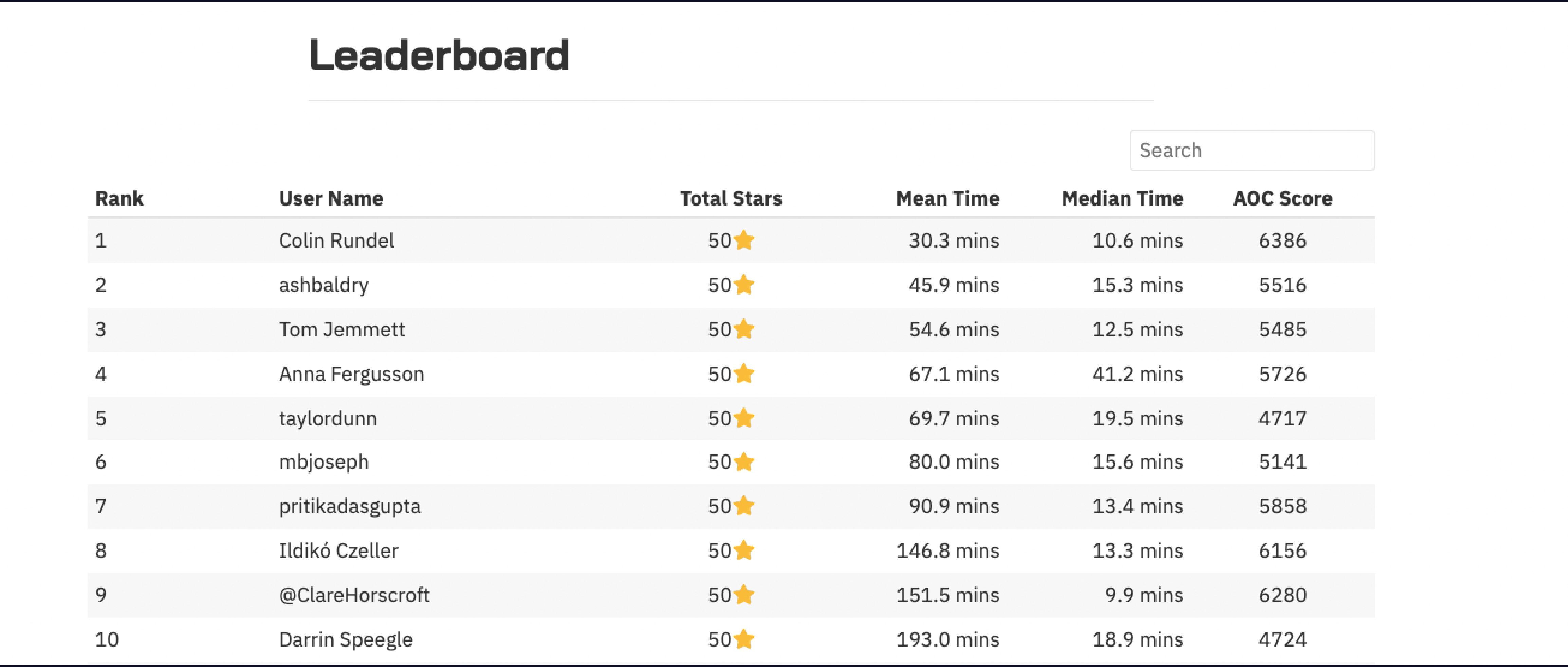
```
First hundred users to get both stars on Day 1:

  1) Dec 01  00:01:07    Tanner Hoke
  2) Dec 01  00:01:12    Erik Amirell Eklöf
  3) Dec 01  00:01:13    David Robinson
  4) Dec 01  00:01:14    bjebert
  5) Dec 01  00:01:18    Tim Vermeulen (AoC++)
  6) Dec 01  00:01:26    dan-simon
  7) Dec 01  00:01:29    Oliver Ni (AoC++)
  8) Dec 01  00:01:30    Shaan Keole
  9) Dec 01  00:01:32    tckmn
 10) Dec 01  00:01:33    Dario Sučić (AoC++)
```

go at your own phase



Gold indicates the user got both stars for that day, silver means just the first star, and gray means none.

```
                 1111111111222222
        1234567890123456789012345
  1) 3836 ****************           Emil Hvitfeldt
  2) 3798 ****************           Colin Rundel
  3) 3798 ****************           David Robinson
  4) 3759 ****************           trang1618
  5) 3758 ****************           @ClareHorscroft
  6) 3713 ****************           @_TanHo (AoC++)
  7) 3594 ****************           Ildikó Czeller (AoC++)
  8) 3533 ****************           dhimmel
  9) 3522 ****************           Jarosław Nirski
 10) 3422 ****************           Jonathan Spring (AoC++)
 11) 3409 ****************           pritikadasgupta
 12) 3285 ****************           Josh Gray
 13) 3234 ****************           Anna Fergusson
 14) 3200 ****************           Jean-Rubin
 15) 3184 ****************           gpecci
 16) 3140 ****************           Melinda Tang
 17) 3023 ****************           ashbaldry
 18) 3012 ****************           Tom Jemmett (AoC++)
 19) 2984 ****************           Riinu Pius (AoC++)
 20) 2866 ****************           mbjoseph
```



Leaderboard

Search

Rank	User Name	Total Stars	Mean Time	Median Time	AOC Score
1	Colin Rundel	50⭐	30.3 mins	10.6 mins	6386
2	ashbaldry	50⭐	45.9 mins	15.3 mins	5516
3	Tom Jemmett	50⭐	54.6 mins	12.5 mins	5485
4	Anna Fergusson	50⭐	67.1 mins	41.2 mins	5726
5	taylordunn	50⭐	69.7 mins	19.5 mins	4717
6	mbjoseph	50⭐	80.0 mins	15.6 mins	5141
7	pritikadasgupta	50⭐	90.9 mins	13.4 mins	5858
8	Ildikó Czeller	50⭐	146.8 mins	13.3 mins	6156
9	@ClareHorscroft	50⭐	151.5 mins	9.9 mins	6280
10	Darrin Speegle	50⭐	193.0 mins	18.9 mins	4724

The R community is awesome!

Many different problems

- Squares With Three Sides (2016 day 3)
- Encoding Error (2020 day 9)
- Wizard Simulator 20XX (2015 day 22)

the problems are CLEARLY written

Take home

- You might learn something
- Join the Community
- Have fun!