

[Advent of Code](#) [\[About\]](#) [\[Events\]](#) [\[Shop\]](#) [\[Settings\]](#) [\[Log Out\]](#) [hyattj-osu 8*](#)
[{:year 2020}](#) [\[Calendar\]](#) [\[AoC++\]](#) [\[Sponsors\]](#) [\[Leaderboard\]](#) [\[Stats\]](#)

--- Day 4: Passport Processing ---

You arrive at the airport only to realize that you grabbed your North Pole Credentials instead of your passport. While these documents are extremely similar, North Pole Credentials aren't issued by a country and therefore aren't actually valid documentation for travel in most of the world.

It seems like you're not the only one having problems, though; a very long line has formed for the automatic passport scanners, and the delay could upset your travel itinerary.

Due to some questionable network security, you realize you might be able to solve both of these problems at the same time.

The automatic passport scanners are slow because they're having trouble detecting which passports have all required fields. The expected fields are as follows:

- `byr` (Birth Year)
- `iyr` (Issue Year)
- `eyr` (Expiration Year)
- `hgt` (Height)
- `hcl` (Hair Color)
- `ecr` (Eye Color)
- `pid` (Passport ID)
- `cid` (Country ID)

Passport data is validated in batch files (your puzzle input). Each passport is represented as a sequence of `key:value` pairs separated by spaces or newlines. Passports are separated by blank lines.

Here is an example batch file containing four passports:

```
ecl:gry pid:860033327 eyr:2020 hcl:#ffffffd
byr:1937 iyr:2017 cid:147 hgt:183cm

iyr:2013 ecl:amb cid:350 eyr:2023 pid:028048884
hcl:#cfa07d byr:1929

hcl:#ae17e1 iyr:2013
eyr:2024
ecl:brn pid:760753108 byr:1931
hgt:179cm

hcl:#cfa07d eyr:2025 pid:166559648
iyr:2011 ecl:brn hgt:59in
```

The first passport is valid - all eight fields are present. The second passport is invalid - it is missing `hgt` (the Height field).

The third passport is interesting; the only missing field is `cid`, so it looks like data from North Pole Credentials, not a passport at all! Surely, nobody would mind if you made the system temporarily ignore missing `cid` fields. Treat this "passport" as valid.

The fourth passport is missing two fields, `cid` and `byr`. Missing `cid` is fine, but missing any other field is not, so this passport is invalid.

According to the above rules, your improved system would report `2` valid passports.

Our [sponsors](#) help make Advent of Code possible:

[Expedition Tech](#) - What will you do when Santa gives you a black box? Look inside.

Count the number of valid passports - those that have all required fields. Treat `cid` as optional. In your batch file, how many passports are valid?

Your puzzle answer was `196`.

--- Part Two ---

The line is moving more quickly now, but you overhear airport security talking about how passports with invalid data are getting through. Better add some data validation, quick!

You can continue to ignore the `cid` field, but each other field has strict rules about what values are valid for automatic validation:

- `byr` (Birth Year) - four digits; at least `1920` and at most `2002`.
- `iyr` (Issue Year) - four digits; at least `2010` and at most `2020`.
- `eyr` (Expiration Year) - four digits; at least `2020` and at most `2030`.
- `hgt` (Height) - a number followed by either `cm` or `in`:
 - If `cm`, the number must be at least `150` and at most `193`.
 - If `in`, the number must be at least `59` and at most `76`.
- `hcl` (Hair Color) - a `#` followed by exactly six characters `0-9` or `a-f`.
- `ekl` (Eye Color) - exactly one of: `amb` `blu` `brn` `gry` `grn` `hzl` `oth`.
- `pid` (Passport ID) - a nine-digit number, including leading zeroes.
- `cid` (Country ID) - ignored, missing or not.

Your job is to count the passports where all required fields are both present and valid according to the above rules. Here are some example values:

```
byr valid: 2002
byr invalid: 2003

hgt valid: 60in
hgt valid: 190cm
hgt invalid: 190in
hgt invalid: 190

hcl valid: #123abc
hcl invalid: #123abz
hcl invalid: 123abc

ekl valid: brn
ekl invalid: wat

pid valid: 000000001
pid invalid: 0123456789
```

Here are some invalid passports:

```
eyr:1972 cid:100
hcl:#18171d ekl:amb hgt:170 pid:186cm iyr:2018 byr:1926

iyr:2019
hcl:#602927 eyr:1967 hgt:170cm
ekl:grn pid:012533040 byr:1946

hcl:dab227 iyr:2012
ekl:brn hgt:182cm pid:021572410 eyr:2020 byr:1992 cid:277

hgt:59cm ekl:zzz
eyr:2038 hcl:74454a iyr:2023
pid:3556412378 byr:2007
```

Here are some valid passports:

```
pid:087499704 hgt:74in ecl:grn iyr:2012 eyr:2030 byr:1980  
hcl:#623a2f  
  
eyr:2029 ecl:blu cid:129 byr:1989  
iyr:2014 pid:896056539 hcl:#a97842 hgt:165cm  
  
hcl:#888785  
hgt:164cm byr:2001 iyr:2015 cid:88  
pid:545766238 ecl:hzl  
eyr:2022  
  
iyr:2010 hgt:158cm hcl:#b6652a ecl:blu byr:1944 eyr:2021 pid:093154719
```

Count the number of valid passports - those that have all required fields and valid values. Continue to treat `cid` as optional. In your batch file, how many passports are valid?

Your puzzle answer was `114`.

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.

If you still want to see it, you can [get your puzzle input](#).

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