

```

In[1]:= rawPassports = ReadString["F:\\prog\\aoc2020\\04input.txt"];
passports =
  StringSplit[#, {"\n", " "}] & /@ StringSplit[#, "\n\n"] & @rawPassports;
requiredFields = {"byr", "iyr", "eyr", "hgt", "hcl", "ecl", "pid"};

In[4]:= validator[passport_] := (
  keys = StringTake[passport, 3];
  Return@SubsetQ[keys, requiredFields]
)

In[5]:= Plus @@ Boole /@ validator /@ passports
Out[5]= 213

In[6]:= hclRef = Table[ToString[n], {n, 0, 9}] ~Join~ {"a", "b", "c", "d", "e", "f"};
eclRef = {"amb", "blu", "brn", "gry", "grn", "hzl", "oth"};
pidRef = Table[ToString[n], {n, 0, 9}];

checkKey[key_, val_] := (
  If[key == "byr", Return[1920 ≤ ToExpression@val ≤ 2002]];
  If[key == "iyr", Return[2010 ≤ ToExpression@val ≤ 2020]];
  If[key == "eyr", Return[2020 ≤ ToExpression@val ≤ 2030]];
  If[key == "hgt", Return[If[StringTake[val, -2] == "cm",
    150 ≤ ToExpression@StringDrop[val, -2] ≤ 193, If[StringTake[val, -2] == "in",
    59 ≤ ToExpression@StringDrop[val, -2] ≤ 76, False]]]];
  If[key == "hcl", Return[StringTake[#, 1] & @val == "#" &&
    SubsetQ[hclRef, StringSplit[#, ""] & @StringDrop[val, 1]]]];
  If[key == "ecl", Return[SubsetQ[eclRef, {val}]]];
  If[key == "pid",
    Return[StringLength[val] == 9 && SubsetQ[pidRef, StringSplit[val, "]]]];
  If[key == "cid", Return[True]];
  Return[False]
)

newValidator[passport_] := (
  keys = StringTake[passport, 3];
  vals = StringDrop[passport, 4];
  oldValidity = validator@passport;
  newValidity = And @@ MapThread[checkKey, {keys, vals}];
  Return[oldValidity && newValidity]
)

In[11]:= Plus @@ Boole /@ newValidator /@ passports
Out[11]= 147

```