

[Advent of Code](#)
[\[About\]](#)
[\[Events\]](#)
[\[Shop\]](#)
[\[Settings\]](#)
[\[Log Out\]](#)
 Jan Roubos (Sponsor) 23\*  
 0.0.0.0:2021
 [\[Calendar\]](#)
[\[AoC++\]](#)
[\[Sponsors\]](#)
[\[Leaderboard\]](#)
[\[Stats\]](#)

--- Day 14: Extended Polymerization ---

The incredible pressures at this depth are starting to put a strain on your submarine. The submarine has `polymerization` equipment that would produce suitable materials to reinforce the submarine, and the nearby volcanically-active caves should even have the necessary input elements in sufficient quantities.

The submarine manual contains instructions for finding the optimal polymer formula; specifically, it offers a polymer template and a list of pair insertion rules (your puzzle input). You just need to work out what polymer would result after repeating the pair insertion process a few times.

For example:

NNCB
CH -> B
HH -> N
CB -> H
NH -> C
HB -> C
HC -> B
HN -> C
NN -> C
BH -> H
NC -> B
NB -> B
BN -> B
BB -> N
BC -> B
CC -> N
CN -> C

The first line is the polymer template - this is the starting point of the process.

The following section defines the pair insertion rules. A rule like `AB -> C` means that when elements `A` and `B` are immediately adjacent, element `C` should be inserted between them. These insertions all happen simultaneously.

So, starting with the polymer template `NNCB`, the first step simultaneously considers all three pairs:

- The first pair (`NN`) matches the rule `NN -> C`, so element `C` is inserted between the first `N` and the second `N`.
- The second pair (`NC`) matches the rule `NC -> B`, so element `B` is inserted between the `N` and the `C`.
- The third pair (`CB`) matches the rule `CB -> H`, so element `H` is inserted between the `C` and the `B`.

Note that these pairs overlap: the second element of one pair is the first element of the next pair. Also, because all pairs are considered simultaneously, inserted elements are not considered to be part of a pair until the next step.

After the first step of this process, the polymer becomes `NCNBCHB`.

Here are the results of a few steps using the above rules:

Template:	NNCB
After step 1:	NCNBCHB
After step 2:	NBCCNBBBCBHCB
After step 3:	NBBBCNCCNBBNBNBBCHBHHBCHB
After step 4:	NBBNBNBBCCNBCNCCNBBNBBNBBNBBBCBHCBHHNHCBBBCBHCB

This polymer grows quickly. After step 5, it has length 97; After step 10, it has length 3073. After step 10, `B` occurs 1749 times, `C` occurs 298 times,

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

[Cerbos](#) - Don't reinvent user permissions!

**H** occurs 161 times, and **N** occurs 865 times; taking the quantity of the most common element (**B**, 1749) and subtracting the quantity of the least common element (**H**, 161) produces  $1749 - 161 = 1588$ .

Apply 10 steps of pair insertion to the polymer template and find the most and least common elements in the result. What do you get if you take the quantity of the most common element and subtract the quantity of the least common element?

Your puzzle answer was **3306**.

The first half of this puzzle is complete! It provides one gold star: ★

--- Part Two ---

The resulting polymer isn't nearly strong enough to reinforce the submarine. You'll need to run more steps of the pair insertion process; a total of 40 steps should do it.

In the above example, the most common element is **B** (occurring **2192039569602** times) and the least common element is **H** (occurring **3849876073** times); subtracting these produces **2188189693529**.

Apply 40 steps of pair insertion to the polymer template and find the most and least common elements in the result. What do you get if you take the quantity of the most common element and subtract the quantity of the least common element?

Answer:  [\[Submit\]](#)

Although it hasn't changed, you can still [get your puzzle input](#).

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