🎄 Advent of Code 2024

Day 2

Part 1

First, read the program.

```
xmul(2,4)&mul[3,7]!^don't()_mul(5,5)+mul(32,64](mul(11,8)undo()?mul(8,5))
```

Then, use the regex `mul\((\d+),(\d+)\)` to find all valid `mul` sequences.

```
(
  (
    start: 1,
    end: 9,
    text: "mul(2,4)",
    captures: ("2", "4"),
  ),
  (
    start: 28,
    end: 36,
    text: "mul(5,5)",
    captures: ("5", "5"),
  ),
  (
    start: 48,
    end: 57,
    text: "mul(11,8)",
    captures: ("11", "8"),
  ),
  (
    start: 64,
    end: 72,
    text: "mul(8,5)",
    captures: ("8", "5"),
  ),
)
```

By following all instructions, we get the following results:

```
(8, 25, 88, 40)
```

Which in sum total to **161**.

Part 2

First, read the program.

```
xmul(2,4)&mul[3,7]!^don't()_mul(5,5)+mul(32,64](mul(11,8)undo()?mul(8,5))
```

Then, use the expanded regex `mul\((\d+),(\d+)\)|do\(\)|don't\(\)` to find all valid `mul` sequences, as well as all `do()` and `don't()` statements.

```
(
  (
    start: 1,
    end: 9,
    text: "mul(2,4)",
    captures: ("2", "4"),
  ),
  (
    start: 20,
    end: 27,
    text: "don't()",
    captures: (none, none),
  ),
  (
    start: 28,
    end: 36,
    text: "mul(5,5)",
    captures: ("5", "5"),
  ),
  (
    start: 48,
    end: 57,
    text: "mul(11,8)",
    captures: ("11", "8"),
  ),
  (
    start: 59,
    end: 63,
    text: "do()",
    captures: (none, none),
  ),
  (
    start: 64,
    end: 72,
    text: "mul(8,5)",
    captures: ("8", "5"),
  ),
)
```

By following all instructions, we get the following results:

```
(8, 40)
```

Which in sum total to **48**.