--- Day 3: Mull It Over ---

--- 第 3 天：仔细考虑---

"Our computers are having issues, so I have no idea if we have any Chief Historians in stock! You're welcome to check the warehouse, though," says the mildly flustered shopkeeper at the North Pole Toboggan Rental Shop. The Historians head out to take a look.

“我们的电脑出了问题，所以我不知道我们是否有首席历史学家的库存！不过，欢迎你检查仓库，“北极雪橇租赁店的店主略显慌乱地说。历史学家们出去看看。

The shopkeeper turns to you. "Any chance you can see why our computers are having issues again?"

店主转向你。“您有机会了解为什么我们的计算机再次出现问题吗？”

The computer appears to be trying to run a program, but its memory (your puzzle input) is corrupted. All of the instructions have been jumbled up!

计算机似乎正在尝试运行一个程序，但它的内存（您的拼图输入）已损坏。所有的说明都被弄乱了！

It seems like the goal of the program is just to multiply some numbers. It does that with instructions like `mul(X,Y)`, where `X` and `Y` are each 1-3 digit numbers. For instance, `mul(44,46)` multiplies `44` by `46` to get a result of `2024`. Similarly, `mul(123,4)` would multiply `123` by `4`.

该程序的目标似乎只是将一些数字相乘。它通过像 `mul（X，Y）` 这样的指令来实现这一点，其中 `X` 和 `Y` 都是 1-3 位数字。例如，`mul（44,46）` 将 `44` 乘以 `46` 得到 `2024` 的结果。同样，`mul（123,4）` 会将 `123` 乘以 `4`。

However, because the program's memory has been corrupted, there are also many invalid characters that should be ignored, even if they look like part of a `mul` instruction. Sequences like `mul(4*`, `mul(6,9!`, `?(12,34)`, or

Our sponsors help make Advent of Code possible:

我们的赞助商帮助使 Advent of Code 成为可能：

Criteo - Helping Santa prepare for the peak season since 2005

Criteo - 自 2005 年以来帮助圣诞老人为旺季做准备

`mul ( 2 , 4 )` do nothing.

但是，由于程序的内存已损坏，因此也有许多无效字符应被忽略，即使它们看起来像 `mul` 指令的一部分。像 `mul(4*`， `mul(6,9!`， `?(12,34)` 或 `mul ( 2 , 4 )` 不执行任何操作。

For example, consider the following section of corrupted memory:

例如，请考虑以下损坏的内存部分：

```
xmul(2,4)%&mul[3,7]!@^do_not_mul(5,5)+mul(32,64]then(mul(11,8)mul(8,5))
```

Only the four highlighted sections are real `mul` instructions. Adding up the result of each instruction produces `161` (`2*4 + 5*5 + 11*8 + 8*5`).

只有四个突出显示的部分是真正的 `mul` 说明。每条指令的结果相加得到 `161` （`2*4 + 5*5 + 11*8 + 8*5`）。

Scan the corrupted memory for uncorrupted `mul` instructions. What do you get if you add up all of the results of the multiplications?

扫描损坏的内存以查找未损坏的 `mul` 指令。如果你把所有乘法的结果加起来，你会得到什么？

Your puzzle answer was `161085926`.

你的谜题答案是 `161085926`。

--- Part Two --- --- 第二部分 ---

As you scan through the corrupted memory, you notice that some of the conditional statements are also still intact. If you handle some of the uncorrupted conditional statements in the program, you might be able to get an even more accurate result.

当您扫描损坏的内存时，您会注意到一些条件语句也仍然完好无损。如果您在程序中处理一些未损坏的条件语句，则可能能够获得更准确的结果。

There are two new instructions you'll need to handle:

您需要处理两个新指令：

- The `do()` instruction enables future `mul` instructions.

  `do ()` 指令启用将来的 `mul` 指令。
- The `don't()` instruction disables future `mul` instructions.

  `don't ()` 指令禁用 future `mul` 指令。

Only the most recent `do()` or `don't()` instruction applies. At the beginning of the program, `mul` instructions are enabled.

只有最新的`do ()` 或 `don't ()` 指令适用。在程序开始时，将启用 `mul` 指令。

For example: 例如：

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

This corrupted memory is similar to the example from before, but this time the `mul(5,5)` and `mul(11,8)` instructions are disabled because there is a `don't()` instruction before them. The other `mul` instructions function normally, including the one at the end that gets re-enabled by a `do()` instruction.

这个损坏的内存与之前的示例类似，但这次 `mul (5,5)` 和 `mul (11,8)` 指令被禁用，因为它们之前有一个 `don't ()` 指令。其他 `mul` 指令正常运行，包括末尾由 `do ()` 指令重新启用的指令。

This time, the sum of the results is `48` (`2*4 + 8*5`).

这一次，结果之和为 `48` （`2*4 + 8*5`）。

Handle the new instructions; what do you get if you add up all of the results of just the enabled multiplications?

处理新指令;如果将启用的乘法的所有结果相加，您会得到什么？

Your puzzle answer was $\boxed{82045421}$.

你的谜题答案是 $\boxed{82045421}$。

Both parts of this puzzle are complete! They provide two gold stars: **

这个拼图的两个部分都是完整的！他们提供两颗金星：**

At this point, you should return to your Advent calendar and try another puzzle.

此时，您应该返回您的降临节日历并尝试另一个拼图。

If you still want to see it, you can get your puzzle input.

如果您仍然想看到它，您可以获取您的拼图输入。

You can also [Share] this puzzle.

你也可以 [分享] 这个拼图。