Intro

Hello! In this survey, you will be asked to **evaluate the outputs of a Large Language Model** (like ChatGPT) in figuring out the **family relationship** between two people in a story.

These problems are tricky to solve! So, the Language Model **breaks down the problem into subquestions** and **generates code** to answer each of the subquestions.

This is an example of a Language Model's correct output to answer the following question:

Question:

[Gabrielle] has a grandson who is [Kevin]. [Dan] went to his brother [Kevin]'s Birthday party. How is [Dan] related to [Gabrielle]?

Correct Answer = grandson
Language Model's Answer = grandson

Reasoning Chain:

```
# 1. How is [Dan] related to [Kevin]?
relation(Dan, Kevin) = brother
# 2. How is [Kevin] related to [Gabrielle]?
relation(Kevin, Gabrielle) = grandson
# 3. How is [Dan] related to [Gabrielle]?
relation(Dan, Gabrielle)
= relation(Dan, Kevin) @ relation(Kevin, Gabrielle
= brother @ grandson
= grandson
```

A couple of important things to note:

1. `relation(Dan, Kevin)` means **"Dan is Kevin's brother"**, **NOT** the other way around.
2. `brother @ grandson` means **"the brother of one's grandson"**, which is still one's grandson.

Sometimes Language Models happen to get the correct answer, but with a wrong reasoning chain. We want your help in understanding why this is the case!

We've identified two main types of errors that Language Models make when getting the **correct answer** with **the wrong reasoning chain**:

1. Wrong/missing **subquestions**

2. Wrong **code**

Here are some examples:

Question:

[Gabrielle] has a grandson who is [Kevin]. [Dan] went to his brother [Kevin]'s Birthday party. How is [Dan] related to [Gabrielle]?

Correct Answer = grandson
Language Model's Answer = grandson

Wrong Reasoning Chain: Wrong/missing subquestions

```
# 1. How is [Dan] related to [Kevin]?
relation(Dan, Kevin) = brother
# 2. How is [Kevin] related to [Gabrielle]?
relation(Kevin, Gabrielle) = grandson
# 3. How is [Kevin] related to [Gabrielle]?
relation(Kevin, Gabrielle)
= relation(Kevin, Gabrielle)
= grandson
```

The original question is asking about the relationship between [Dan] and [Gabrielle], but the subquestion of "*How is [Dan] related to [Gabrielle]?*" is missing. The answer still happens to be correct, since Kevin is also Gabrielle's grandson.

Wrong Reasoning Chain: Wrong Code

```
# 1. How is [Dan] related to [Kevin]?
relation(Dan, Kevin) = brother
# 2. How is [Kevin] related to [Gabrielle]?
relation(Kevin, Gabrielle) = grandson
# 3. How is [Dan] related to [Gabrielle]?
relation(Dan, Gabrielle)
= relation(Kevin, Gabrielle)
= grandson
```

The model incorrectly assumes that **`relation(Dan, Gabrielle)`** is the same as **`relation(Kevin, Gabrielle)`**, but the answer still happens to be correct.

Now, you are going to see **10** examples like this. Each example has a **family relationship question** and the response generated by a Language Model (LM). The LM's answer is always "correct", and your task is to check whether the **reasoning chain** is also **correct** or **contains any error** (e.g., wrong/missing subquestion, or wrong code). It should take around 15 minutes in total.

Things to note:
1. If you select options other than "reasoning chain is correct", you also need to **justify** your answer briefly.
2. You can assume that the **rules to derive the resulting relationship from two relationships** are **always correct**

(e.g. `brother @ grandson = grandson`).

3. In the middle, you will see one **attention check question**, which will be identical to the example we showed on the previous page. **If you fail the attention check, we will reject the payment**. So please make sure you understand the example. If you need to view these instructions again, click the back button on the bottom left.

4. If you are truly unable to complete the task **even after** searching, you can choose "I'm confused" (however, **too many** such answers may lead to the **rejection** of your response).

Do you understand the task?

○ Yes

○ No (please exit the survey in this case)

Question:

[Gabrielle] has a grandson who is [Kevin]. [Dan] went to his brother [Kevin]'s Birthday party. How is [Dan] related to [Gabrielle]?

Correct Answer = grandson
Language Model's Answer = grandson

Reasoning Chain:

```
# 1. How is [Dan] related to [Kevin]?
relation(Dan, Kevin) = brother
# 2. How is [Kevin] related to [Gabrielle]?
relation(Kevin, Gabrielle) = grandson
# 3. How is [Kevin] related to [Gabrielle]?
relation(Kevin, Gabrielle)
= relation(Kevin, Gabrielle)
= grandson
```

☐ Reasoning chain is correct

☐ Subquestions are incorrect or missing

☐ Generated code is incorrect

☐ There is something wrong with the question itself

Logic 0

Question:

[Jason] and his wife [Gabrielle] baked a cake for [Lisa], his daughter. Question: How is [Lisa] related to [Gabrielle]?

Correct Answer = daughter
Language Model's Answer = daughter

Reasoning Chain:

```
#1. How is [Lisa] related to [Jason]?
relation(Lisa, Jason) = daughter
# 2. How is [Jason] related to [Gabrielle]?
```