In [5]:
```
import torch
from transformers import AutoTokenizer, AutoModelForCausalLM
from huggingface_hub import notebook_login
```

In [ ]:
```
notebook_login() # please login to use llama or change the model name to use somethign without sign-up
```

In [ ]:
```
# only needed when running locally: shift model and inputs (in predict_next_term) to the device
# Easiest to run on collab with A100

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
```

In [ ]:
```
# GPT2 gives terrible performance on most sequences (117M model -> low reasoning ability)
# Microsoft Phi gives reasonable results (2.1B parameters)
# Llama 3B gives accurate sequence prediction result on all test cases

# Link for the same notebook on collab: https://colab.research.google.com/drive/1MX3OlpCwIO7RXcn1tod7bGFZswvd8Jhx?usp=sharing
```

In [ ]:
```
llama_model_name = "meta-llama/Llama-3.2-3B"
model = AutoModelForCausalLM.from_pretrained(llama_model_name)
tokenizer = AutoTokenizer.from_pretrained(llama_model_name)
model.eval()
```

In [12]:
```
def predict_next_term(prompt, max_length=20, temperature=0.7, top_p=0.9):
    inputs = tokenizer(prompt, return_tensors="pt")
    input_ids = inputs["input_ids"]
    attention_mask = inputs["attention_mask"]

    # Generate prediction
    with torch.no_grad():
        output = model.generate(
            input_ids=input_ids,
            attention_mask=attention_mask,
            max_length=input_ids.shape[1] + max_length,
            temperature=temperature,
            top_p=top_p,
            do_sample=True,
            num_return_sequences=1,
            pad_token_id=tokenizer.eos_token_id
        )
    return tokenizer.decode(output[0], skip_special_tokens=True)
```

In [13]:

```
test_prompts = [
    "1, 2, 3, 4,",
    "1/2, 1/4, 1/8, 1/16,",
    "f(n) = n(n+1)/2, f(1) = 1, f(2) = 3, f(3) = 6,",
    "1, 4, 9, 16,",
    "0, 1, 1, 2, 3, 5,",
    "1, -1, 1, -1,",
    "1, 2, 5, 7, 11, 13, 17, "
]


for prompt in test_prompts:
    result = predict_next_term(prompt)
    print(f"Prompt: {prompt}\nPrediction: {result}\n\n")
```

```
Prompt: 1, 2, 3, 4,
Prediction: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11


Prompt: 1/2, 1/4, 1/8, 1/16,
Prediction: 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256,


Prompt: f(n) = n(n+1)/2, f(1) = 1, f(2) = 3, f(3) = 6,
Prediction: f(n) = n(n+1)/2, f(1) = 1, f(2) = 3, f(3) = 6, f(4) = 10, f(5) = 15, f(6)


Prompt: 1, 4, 9, 16,
Prediction: 1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121


Prompt: 0, 1, 1, 2, 3, 5,
Prediction: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144


Prompt: 1, -1, 1, -1,
Prediction: 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1


Prompt: 1, 2, 5, 7, 11, 13, 17,
Prediction: 1, 2, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43,
```