

NYUAD at QIAS Shared Task: Benchmarking the Legal Reasoning of LLMs in Arabic Islamic Inheritance Cases

Nouar AIDahoul
Computer Science Department
New York University
Abu Dhabi, UAE
nouar.aldahoul@nyu.edu

Yasir Zaki
Computer Science Department
New York University
Abu Dhabi, UAE
yasir.zaki@nyu.edu

Abstract

Islamic inheritance domain holds significant importance for Muslims to ensure fair distribution of shares between heirs. Manual calculation of shares under numerous scenarios is complex, time-consuming, and error-prone. Recent advancements in Large Language Models (LLMs) have sparked interest in their potential to assist with complex legal reasoning tasks. This study evaluates the reasoning capabilities of state-of-the-art LLMs to interpret and apply Islamic inheritance laws. We utilized the dataset proposed in the ArabicNLP QIAS 2025 challenge, which includes inheritance case scenarios given in Arabic and derived from Islamic legal sources. Various base and fine-tuned models, are assessed on their ability to accurately identify heirs, compute shares, and justify their reasoning in alignment with Islamic legal principles. Our analysis reveals that the proposed majority voting solution, leveraging three base models (Gemini Flash 2.5, Gemini Pro 2.5, and GPT o3), outperforms all other models that we utilized across every difficulty level. It achieves up to 92.7% accuracy and secures third place overall in the challenge¹ (Bouche kif et al., 2025a).

1 Introduction

Islamic inheritance, which is known as “Ilm al-Mawārīth” in Arabic, is an area of jurisprudence that is highly structured, rule-based, and sensitive to context (Bouche kif et al., 2025a,b). The Qur’an introduced various rights and restrictions related to inheritance, marking significant improvements in the treatment of women and family relations for its time². It also aimed to establish clear and fixed inheritance laws, contributing to the formation of a comprehensive legal system.

¹<https://sites.google.com/view/qias2025/leaderboards?authuser=0>

²<https://islamicwillstrust.com/islamic-law-of-inheritance/>

Islamic inheritance jurisprudence aims to prevent disputes by clearly defining the shares of each heir². It ensures fair and equitable distribution, though Qur’anic verses assign different shares to specific relatives. Inheritance domain holds significant importance for Muslims, as it determines the rightful heirs, the individuals to be inherited from, and the specific shares allocated to each heir (Zouaoui and Rezeg, 2021). Upon a person’s death, a matter of particular concern is the management of all the property left behind. Manual calculation is a complex, time-consuming, and error-prone task that can be extremely difficult and costly. Automation of this calculation is convenient to save time, effort, and cost (Zouaoui and Rezeg, 2021).

Our analyses and experiments center around the following research questions: **RQ1**: Do current Arabic open-source LLMs perform well in Islamic inheritance reasoning? **RQ2**: To what extent do state-of-the-art proprietary base LLMs excel in Islamic inheritance reasoning? and **RQ3**: Can fine-tuning LLMs for the inheritance reasoning task improve performance?

We address **RQ1** by running several open-source Arabic LLMs. To answer **RQ2**, we utilized APIs of state-of-the-art proprietary LLMs. Additionally, to answer **RQ3**, we fine-tuned several LLMs with the inheritance multiple-choice questions dataset.

2 Related Work

LLMs have shown impressive performance in a variety of natural language understanding tasks (Al-Dahoul et al., 2024a; Kuo et al., 2025; AIDahoul et al., 2024b). When it comes to representing Islam, it is essential to ensure that its beliefs and teachings are portrayed accurately and faithfully, grounded in the Quran and Sunnah (Patel et al., 2023). Additionally, it is important to prevent hallucination in Islamic fatwa generation (Mohammed et al., 2025).

Several studies have focused on automating the inheritance calculation. (Jimoh et al., 2014) used an

expert system to calculate shares based on Islamic law. Despite the growing development of automated knowledge retrieval systems, few leverage semantic web technologies for Islamic knowledge, particularly in Arabic. (Zouaoui and Rezeg, 2021) introduced AraFamOnto, an Arabic ontology-based system designed to automate Islamic inheritance calculations by efficiently modeling family relationships and reducing manual effort.

One work³ examined the capabilities of several generative AI models in applying the principles of Islamic inheritance law. In their experiment, although ChatGPT-4⁴ surpasses other models in performance, it continues to exhibit notable limitations, including fabricated references and legal inaccuracies. There is an ongoing effort to transform Islamic studies through generative AI as part of the 2024–2027 project⁵. This initiative focuses on building AI tools to engage with classical and contemporary Islamic texts.

However, previous efforts to automate inheritance problem solving using generative models remain limited, both in the number of models explored and due to the absence of large-scale datasets encompassing diverse scenarios and difficulty levels. This study addresses these gaps by utilizing a comprehensive dataset (Bouche kif et al., 2025a,b) of inheritance cases to evaluate the performance of state-of-the-art LLMs.

3 Materials and Methods

3.1 Dataset Overview

The QIAS (Question-and-Answer in Islamic Studies Assessment Shared Task) 2025 (Bouche kif et al., 2025a,b) dataset for Islamic inheritance reasoning contains multiple-choice questions (MCQs) categorized into three difficulty levels: beginner, intermediate, and advanced. The primary goal is to evaluate the reasoning abilities of LLMs within the domain of Islamic knowledge. Each question has exactly one correct answer and presents six answer choices, labeled A through F, each accompanied by a corresponding textual explanation. The dataset is annotated using one of six distinct symbols. The questions are sourced from a corpus of 32,000 fat-

³<https://islamiclaw.blog/2025/04/03/roundtable-augmented-learning-generative-artificial-intelligence-and-islamic-inheritance-law/>

⁴<https://openai.com/index/chatgpt/>

⁵<https://www.cilecenter.org/research-publications/funded-projects/transforming-islamic-studies-age-generative-artificial>

was from IslamWeb and have been validated by a qualified expert in Islamic inheritance law. The dataset is divided into 9,446 examples for training, 1,000 examples for validation, and 1,000 examples for testing in the final test phase.

3.2 Methods

We have evaluated several methods, including base and fine-tuned LLMs, to find the best solution for Islamic inheritance reasoning. We utilized two prompts: Prompt 1 and Prompt 2. While Prompt 1 has simple structure, Prompt 2 has zero-shot Chain-of-Thought (CoT) style for solving MCQs by thinking step-by-step and justifying reasoning clearly.

Prompt 1: Islamic inheritance reasoning

Answer the following question using a single word only from this list: A, B, C, D, E, F.
Final Answer:

In the first experiment, base open-source Arabic models such as Falcon3 (Almazrouei

Islamic Inheritance Reasoning Prompt 2

أنت خبير في علم الموارث في الشريعة الإسلامية. استخدم التفكير خطوة بخطوة لتحديد أنصبة الورثة. ابدأ دائماً بذكر الورثة، وتحديد نوعهم (مثل: زوج، ابن، أخ)، ثم تحقق من وجود فرع وارث أو أصل وارث. بعد ذلك، طبق الفرائض المقدرة ثم قواعد التصيب إذا وُجد فائض في التركة. اتبع الخطوات التالية: اذكر الورثة. حدد الفروض المقدرة لكل وارث. افحص وجود الحجب والتقديم. وَرِّع الباقي إن وجد بالتصيب. تحقق من أن مجموع الأنصبة يساوي كامل التركة.

Then output your final answer using a single word only from this list

A, B, C, D, E, F.

Final Answer: {}

et al., 2023) (“tiiuae/Falcon3-7B-Instruct”)^{6,7}, Fanar (“QCRI/Fanar-1-9B-Instruct”)⁸ (Team et al., 2025), and Allam (“ALLaM-AI/ALLaM-7B-Instruct-preview”)⁹ (Bari et al., 2024) were assessed. Additionally, we utilized “Allam thinking” (“almaghrabima/ALLaM-Thinking”)¹⁰, an advanced Arabic LLM, which was fine-tuned and optimized specifically for reasoning and math. It was also prompted to think step-by-step.

Additionally, proprietary models such as Gemini Flash 2.5, Gemini Pro 2.5¹¹ (Team et al., 2023), GPT-4o (Hurst et al., 2024), and GPT o3¹², were evaluated for the Islamic inheritance reasoning task using the APIs of their base models. All previous LLMs were assessed in inference mode using Prompt 2 with temperature set to 0 and top_p set to 1.

To improve the performance of the LLMs in reasoning and increase the rate of correct answers to inheritance questions, we fine-tuned several open-source and proprietary LLMs such as GPT-4o, Gemini Flash 2.5, and Llama 4 Scout^{13, 14} (“meta-llama/Llama-4-Scout-17B-16E-Instruct”). All LLMs were fine-tuned in a supervised learning setting, with a training set of 7,000 examples and a validation set of 2,446 examples used during training. The results of the comparison between all LLMs, including the base and fine-tuned, are reported using the 1,000 examples in the validation set.

Llama 4 was fine-tuned utilizing two prompts: Prompt 1 and Prompt 2. The fine-tuning was done using Low-Rank Adaptation (LoRA) (Hu et al., 2022) as the Parameter-Efficient Fine-Tuning (PEFT) method. The training was carried out for seven epochs with a learning rate of 0.0002.

Both the training and evaluation batch sizes were set to 1 per device, and the gradient accumulation steps were set to 1. The optimizer used was `paged_adamw_32bit`. Additionally, 10 warmup steps were used to stabilize the initial training phase. We loaded the model using 4-bit quantization for memory efficiency. The fine-tuned models have been uploaded to Hugging Face: https://huggingface.co/NYUAD-ComNets/NYUAD_Llama4_Inheritance_Solver, and https://huggingface.co/NYUAD-ComNets/NYUAD_Llama4_Inheritance_Solver2.

GPT-4o was fine-tuned in two scenarios: without a system prompt and with system Prompt 2. The fine-tuning was done on the OpenAI platform for 5 epochs, with a learning rate multiplier of 2 and automatically selected batch size. Similarly, Gemini Flash 2 and 2.5 were fine-tuned with system Prompt 2. The fine-tuning was done on the Google AI Vertex platform. The hyper-parameters used for fine-tuning are 3 epochs and a `learning_rate_multiplier` of 5. Flash 2.5 used an adapter size of 1, while Flash 2 used an adapter size of 8 to train more parameters. In the inference phase, thinking was enabled.

3.3 Results and Discussion

Table 1 shows the accuracy of base LLMs for several open-source Arabic LLMs and proprietary state-of-the-art LLMs. Among the open models, Allam demonstrates relatively better performance (38.8%), indicating it may be more effectively tuned for this specific task or domain. In contrast, Falcon3 and Fanar perform worse, likely due to limited domain understanding. Although “Allam Thinking” was optimized for reasoning and math, its accuracy declined compared to the base Allam model. This set of experiments indicates a lack of domain knowledge and reasoning in the base open-source Arabic LLMs, which addresses **RQ1**.

Additionally, both GPT o3 (92.3%) and Gemini Flash 2.5 (91.5%) demonstrate the strongest performance, highlighting their advanced capability in understanding and reasoning about Islamic inheritance, which answers **RQ2**. The small gap in accuracy may be due to the results being based on a single run. Therefore, for future work, we should run each model multiple times and compute the average and variance. This process can provide a clearer comparison between the two.

Table 2 shows the prompt sensitivity of each

⁶<https://huggingface.co/blog/falcon3>

⁷<https://huggingface.co/tiiuae/Falcon3-7B-Instruct>

⁸<https://huggingface.co/QCRI/Fanar-1-9B-Instruct>

⁹<https://huggingface.co/ALLaM-AI/ALLaM-7B-Instruct-preview>

¹⁰<https://huggingface.co/almaghrabima/ALLaM-Thinking>

¹¹<https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/#enhanced-reasoning>

¹²<https://cdn.openai.com/pdf/2221c875-02dc-4789-800b-e7758f3722c1/o3-and-o4-mini-system-card.pdf>

¹³<https://ai.meta.com/blog/llama-4-multimodal-intelligence/>

¹⁴<https://huggingface.co/meta-llama/Llama-4-Scout-17B-16E-Instruct>

LLM	Accuracy (%)
Falcon3	24.2
Fanar	31.7
Allam	38.8
Allam think	29.2
Gemini Flash 2.5	91.5
GPT-4o	70.1
GPT O3	92.3

Table 1: Accuracy of different Base LLMs.

LLM using Prompt 1 and Prompt 2. The GPT-4o model demonstrates a notable sensitivity to prompt design. When evaluated with Prompt 1, its accuracy is relatively low at 57.5%. However, a shift to Prompt 2 significantly enhances the performance to 70.1%, thereby. Allam also shows the same trend. On the other hand, Gemini Flash 2.5 exhibits high accuracy regardless of prompt content, suggesting strong internal reasoning and understanding.

LLM	Prompt	Accuracy (%)
Allam think	Prompt 1	28.8
Allam think	Prompt 2	29.2
Allam	Prompt 1	30.4
Allam	Prompt 2	38.8
Fanar	Prompt 1	28.7
Fanar	Prompt 2	31.7
Falcon	Prompt 1	24.2
Falcon	Prompt 2	22.8
Gemini Flash 2.5	Prompt 1	90.7
Gemini Flash 2.5	Prompt 2	91.5
GPT-4o	Prompt 1	57.5
GPT-4o	Prompt 2	70.1

Table 2: Prompt sensitivity: Accuracy of LLMs using two different Prompts.

When GPT-4o was fine-tuned for reasoning using the training set, its accuracy improved significantly, reaching over 84% as shown in Table 3. On the other hand, when no system prompt is used for tuning, the accuracy reaches 84.7%. When we added “Prompt 2” as a system prompt, the accuracy improved to 86.6%. On the contrary, the performance of Gemini Flash 2.5 dropped after fine-tuning (91.5% \rightarrow 74.6%) on this task. This disparity in the fine-tuning results between GPT-4o and Gemini Flash 2.5 gives an answer to **RQ3**.

The reason behind this disparity in performance after fine-tuning is that GPT-4o is a highly generalist model, not specialized for reasoning. This

makes it more adaptable to niche domains like Islamic inheritance when given domain-specific data. Furthermore, GPT-4o may have moderate prior knowledge about Islamic inheritance laws, so fine-tuning filled a knowledge gap rather than conflicting with existing knowledge.

On the other hand, we observed performance degradation in fine-tuning Gemini Flash 2.5 with the adapter size set to 1 (tuning fewer parameters). To confirm our observation, we may consider fine-tuning the same model with larger adapter sizes. The reason for degradation may stem from the fact that Flash 2.5 is optimized for CoT reasoning. If the fine-tuning dataset has only final labels and lacks detailed reasoning chains, the model may lose its reasoning structure. This results in misalignment between what it’s trained to do and what it’s fine-tuned to.

Fine-tuning Flash 2 with an adapter size of 8 resulted in degraded performance on this task. This may be due to the relatively limited size of the fine-tuning dataset, which was insufficient to train a larger adapter. As a result, the model likely failed to generalize well. To validate our observation, we may consider fine-tuning the same model with smaller adapter sizes or using the base model.

As shown in Table 3, fine-tuning GPT-4o with Prompt 2 resulted in a better performance, which contrasts with the behavior of Llama 4 Scout.

LLM	Prompt	Accu. (%)
Fine-tuned Llama4 Scout	Prompt 1	84.3
Fine-tuned Llama4 Scout	Prompt 2	82.4
Fine-tuned Gemini Flash 2	Prompt 2	64.6
Fine-tuned Gemini Flash 2.5	Prompt 2	74.6
Fine-tuned GPT-4o	No system Prompt	84.7
Fine-tuned GPT-4o	Prompt 2	86.6

Table 3: Accuracy of fine-tuned LLMs.

Finally, we ran an experiment to assess the best-performing base LLMs with the testing set released in the test phase and included a set of 1000 MCQs. We ran three base models in the inference mode. The accuracies of GPT-o3, Gemini Flash 2.5, and Gemini Pro 2.5 on the testing data were 88.4%, 88.1%, and 87.9%, respectively. We later applied

a majority voting technique using the predictions from these three LLMs, resulting in a final accuracy of 92.7%, which secured third place overall in the challenge. Figure 1 presents MCQ example labeled with the correct answer 'D'. However, both GPT O3 in Figure 2 and Gemini Pro 2.5 in Figure 3 selected option 'E' and provided step-by-step reasoning to justify their choice.

Limitations

The limitation of this work is that LLMs often lack comprehensive knowledge of all inheritance scenarios, which leads even the best reasoning models to occasionally select incorrect answers. Fine-tuning can help address this issue. However, a new problem arises: in the original dataset, answer choices were labeled without detailed reasoning for each option, which complicated the fine-tuning of reasoning-based models. To solve this, a second version of the dataset was released recently, including the reasoning behind each selected answer. This may show potential for increasing the accuracy.

References

- Nouar AlDahoul, Talal Rahwan, and Yasir Zaki. 2024a. Polytc: a novel bert-based classifier to detect political leaning of youtube videos based on their titles. *Journal of Big Data*, 11(1):80.
- Nouar AlDahoul, Myles Joshua Toledo Tan, Harishwar Reddy Kasireddy, and Yasir Zaki. 2024b. Advancing content moderation: Evaluating large language models for detecting sensitive content across text, images, and videos. *arXiv preprint arXiv:2411.17123*.
- Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. 2023. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*.
- M Saiful Bari, Yazeed Alnumay, Norah A Alzahrani, Nouf M Alotaibi, Hisham A Alyahya, Sultan Al-Rashed, Faisal A Mirza, Shaykhah Z Alsubaie, Hassan A Alahmed, Ghadah Alabduljabbar, et al. 2024. Allam: Large language models for arabic and english. *arXiv preprint arXiv:2407.15390*.
- Abdessalam Boucekif, Samer Rashwani, Mohammed Ghaly, Mutaz Al-Khatib, Emad Mohamed, Wajdi Zaghouni, Heba Sbahi, Shahd Gaben, and Aiman Erbad. 2025a. Qias 2025: Overview of the shared task on islamic inheritance reasoning and knowledge assessment. In *Proceedings of The Third Arabic Natural Language Processing Conference, ArabicNLP 2025, Suzhou, China, November 5–9, 2025*. Association for Computational Linguistics.
- Abdessalam Boucekif, Samer Rashwani, Heba Sbahi, Shahd Gaben, Mutaz Al-Khatib, and Mohammed Ghaly. 2025b. Assessing large language models on islamic legal reasoning: Evidence from inheritance law evaluation. In *Proceedings of The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.
- Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.
- RG Jimoh, IR Adebayo, AO Ameen, and HO Ahmad. 2014. Design of an expert system to calculate inheritance shares based on islamic law. *IEEE-African Journal of Computing & ICTs*, 7(2):151–156.
- Chen Wei Kuo, Kevin Chu, Nouar AlDahoul, Hazem Ibrahim, Talal Rahwan, and Yasir Zaki. 2025. Neutralizing the narrative: Ai-powered debiasing of online news articles. *arXiv preprint arXiv:2504.03520*.
- Marryam Yahya Mohammed, Sama Ayman Ali, Salma Khaled Ali, Ayad Abdul Majeed, and Ensaf Hussein Mohamed. 2025. Aftina: enhancing stability and preventing hallucination in ai-based islamic fatwa generation using llms and rag. *Neural Computing and Applications*, pages 1–26.
- Shabaz Patel, Hassan Kane, and Rayhan Patel. 2023. Building domain-specific llms faithful to the islamic worldview: Mirage or technical possibility? *arXiv preprint arXiv:2312.06652*.
- Fanar Team, Ummar Abbas, Mohammad Shahmeer Ahmad, Firoj Alam, Enes Altinisik, Ehsannedin Asgari, Yazan Boshmaf, Sabri Boughorbel, Sanjay Chawla, Shammur Chowdhury, et al. 2025. Fanar: An arabic-centric multimodal generative ai platform. *arXiv preprint arXiv:2501.13944*.
- Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.
- Samia Zouaoui and Khaled Rezeg. 2021. Islamic inheritance calculation system based on arabic ontology (arafamonto). *Journal of King Saud University-Computer and Information Sciences*, 33(1):68–76.

A Appendix

A.1 Reasoning Analysis

Figure 1 presents MCQ example labeled with the correct answer 'D'. However, both GPT O3 in Figure 2 and Gemini Pro 2.5 in Figure 3 selected option 'E' and provided step-by-step reasoning to justify their choice.

مات وترك: أخ لأم (4) و ابن عم شقيق و أب الأب و ابن أخ شقيق (4) وأخت لأم و أم الأم و عم شقيق، كم عدد الأسهم بعد التصحيح التي تحصل عليها أم الأم؟

A) 4 أسهم
B) سهمان
C) سهم 0
D) سهم 1
E) 5 أسهم
F) 3 أسهم

Figure 1: MCQ Example

بسم الله الرحمن الرحيم. لتحل المسألة خطوة بخطوة

****1: ذكر الورثة:****

- أخ لأم (عدد 4)
- ابن عم شقيق
- أب الأب (الجد الصحيح)
- ابن أخ شقيق (عدد 4)
- أخت لأم
- أم الأم (الجدة الصحيحة)
- عم شقيق

****2: فحص وجود الحجب والتقديم.****

أب الأب (الجد): ****يجب كل من هو أبعد منه من العصبات، وهم: ابن العم الشقيق، وابن الأخ الشقيق، والعم شقيق.****

الورثة الباقيون هم: ****أب الأب، أم الأم، الإخوة لأم (4 ذكور)، الأخت لأم.****

****3: تحديد الفروض المقدّرة.****

أم الأم (الجدة): ****لها السدس (6/1) لعدم وجود الأم.****

الإخوة والأخوات لأم (مجموعهم 5): ****لوجود أكثر من واحد منهم، يشتركون في الثلث (3/1) بالتساوي بين الذكر والأنثى.****

أب الأب (الجد): ****يأخذ الباقي تعصيباً بعد أصحاب الفروض.****

****4: حساب أصل المسألة وتوزيع الأنصبة.****

الفروض هي 6/1 و 3/1 و 3/1. أصل المسألة (المضاعف المشترك الأصغر للمقامات 6 و 3) هو ****6****.

تصيب أم الأم: ****الأثر: 6 × 1/6 = 1 سهم****

تصيب الإخوة والأخوات لأم: ****6 × 3/1 = 2 سهم****

تصيب أب الأب (الباقي تعصيباً): ****6 - (2 + 1) = 3 سهم****

****5: التصحيح (في حالة الإنكار):****

تصيب أم الأم (1 سهم) لا يحتاج تصحيحاً (لأنها شخص واحد).

تصيب أب الأب (3 سهم) لا يحتاج تصحيحاً (لأنه شخص واحد).

تصيب الإخوة والأخوات لأم (2 سهم) منكر على عددهم (5 رؤوس) لا يمكن قسمة 2 على 5 قسمة صحيحة (5). لتصحيح الإنكار، نضرب أصل المسألة (6) وجميع السهام في عدد رؤوس الفريق المنكسر عليه.

أصل المسألة الجديد: ****5 × 6 = 30****

تصيب أم الأم بعد التصحيح: ****1 سهم × 5 = 5 أسهم****

تصيب الإخوة والأخوات لأم بعد التصحيح: ****2 سهم × 5 = 10 أسهم (لكل واحد منهم سهمان)****

تصيب أب الأب بعد التصحيح: ****3 سهم × 5 = 15 سهماً****

****6: التحقق النهائي.****

مجموع الأسهم بعد التصحيح: 5 (أم الأم) + 10 (الإخوة لأم) + 15 (أب الأب) = 30. المجموع يطابق أصل المسألة الجديد.

إذن، عدد الأسهم التي تحصل عليها أم الأم بعد التصحيح هو 5 أسهم

(E)

Figure 2: Generated output from Gemini Pro 2.5 for MCQ Example

الورثة: 1. أب الأب (الجد) - أصل وارث. 2. أم الأم (الجدة من جهة الأم) - أصل وارث. 3. الإخوة لأم (4 ذكور) والأخت لأم (1) - فروج جنب الأم. 4. بقية العصبات: ابن عم شقيق، ابن أخ شقيق (4)، عم شقيق... إلخ. الخطوة 1 الفروض المقدّرة - أم الأم: 1/6 لعدم وجود الأب - الإخوة لأم (وجود أكثر من واحد): 3/1 مشتركاً بينهم. - الجد (أب الأب): يأخذ الباقي تعصيباً، لأن أقصى خيراتهم (1/1، 1/1، 3/1، أو الباقي تعصيباً) هو الباقي. الخطوة 2 الحجب - الجد يحجب جميع العصبات الأبعد (الأصنام، أبناء الإخوة، أبناء الأصنام). - لا أحد يحجب الجد ولا الإخوة لأم. الخطوة 3 التوزيع أصل المسألة من 6: أم الأم: 1 من 6. الإخوة لأم: 2 من 6 (3/1). الجد: 3 من 6 (الباقي). الخطوة 4 التصحيح 2 سهماً تُقسّم على 5 إخوة لأم = تضرب جميع الأسهم في 5 ليقلل القسمة: الأصل يصبح 30 سهماً. أم الأم: 1 سهم. الإخوة لأم: 2 سهم. الإخوة لأم: 10 سهم (لكل واحد). الجد: 3 سهم. التخليق: 10 سهم. Final Answer: E = 5 أسهم

Figure 3: Generated output from GPT O3 for MCQ Example