

Projekt z przedmiotu Systemy Automatycznej Identyfikacji

1. Zadanie do wykonania

AI - wykonanie zadania (Znalezienie w pamięci Mifare tekstu, odczyt danych)

- porównanie jakości rozwiązań w kilku modelach AI,
- sprawdzenie, czy model potrafi samodzielnie zweryfikować poprawność rozwiązania,
- ocena, czy model potrafi przewidzieć wynik działania,
- analiza kompletności i czytelności odpowiedzi,
- wskazanie modelu najlepiej radzącego sobie z zadaniem.

2. Sposób wykonania zadania

Testowane modele to najnowsze i najpopularniejsze rozwiązania, w najwyższych wersjach, czyli takie z których większość korzysta na co dzień w ramach np. firmowych subskrypcji. Możliwość przetestowania takich modeli, dla osób nie posiadających wykupionych subskrypcji jest oferowana na stronie [Arena.ai](https://arena.ai), wykorzystanej przeze mnie w trybie tekstowego porównania modeli side-by-side.

3. Prompt

Zadanie wykonano poprzez wpisanie podanego poniżej prompta z wyszczegółowionym opisem roli, celu i instrukcji do wykonania zadania

Jesteś asystentem pomocnym w zadaniach z kart zbliżeniowych MIFARE i systemu kontroli dostępu. Poniżej przesyłam instrukcję i wymagania dotyczące zadania z odczytu pamięci karty MIFARE przy użyciu czytnika ACR122U.

Instrukcja odczytu pamięci MIFARE:

1. Uruchom program: ACR122U\Application\ACR122UTool.exe
2. Reader Commands → New connection
3. Reader Commands → Get ATR
4. Card Commands → Load Authentication Keys
 - Key number → 00
 - Key → FF FF FF FF FF FF
5. Card Commands → Authenticate
 - Block No. → 0 (lub inny blok, który chcemy odczytać)
 - Key Type → Type A
 - Key No. → 0

6. Card Commands → Read Block

- Block numer → 00 (lub inny blok do odczytu)

7. Znalezienie w pamięci MIFare tekstu „LabPKE”

Użyj powyższych materiałów, aby wykonać zadanie:

1. Wyjaśnij krok po kroku, jak odczytać zawartość pamięci karty MIFARE przy użyciu ACR122UTool.exe
2. Opisz, jak znaleźć tekst „LabPKE” w pamięci karty
3. Wyjaśnij różnicę między notacją szesnastkową a dziesiętną w kontekście numeracji bloków
4. Pokaż, jak przeliczyć numer bloku z notacji ciągłej na standard MIFARE
5. Podaj przykład przeliczenia dla bloku 9 (hex)

Wykonaj samoweryfikację podanego rozwiązania:

1. Sprawdź, czy Twoje rozwiązanie jest poprawne:
 - Czy UID jest podany w formacie 4 bajty szesnastkowo?
 - Czy numer bloku jest podany w notacji szesnastkowej?
 - Czy przeliczenie sektora i bloku jest zgodne ze wzorem?
 - Czy odpowiedź jest kompletna (UID, blok hex, sektor dec, blok dec)?
2. Jeśli widzisz błąd, popraw go i wyjaśnij, co było źle.

Wykonaj predykcję wyniku:

1. Przewidź, jaki wynik otrzyma użytkownik, jeśli:
 - Blok w ACRTTool to 0A (hex) = 10 (dec)
 - Oblicz: sektor = ? , blok = ?

4. Odpowiedzi uzyskane od modeli AI

1) Gemini 3.1 pro

Odpowiedź modelu
Witaj! Jako asystent ds. systemów kontroli dostępu i kart MIFARE, chętnie pomogę Ci w tym zadaniu. Poniżej znajduje się kompleksowe rozwiązanie, oparte na dostarczonej instrukcji oraz technicznej specyfikacji kart z rodziny MIFARE (zakładamy standardową kartę MIFARE Classic 1K, która jest najpopularniejsza).

CZĘŚĆ 1: Realizacja zadań z instrukcji

1. Wyjaśnienie krok po kroku, jak odczytać zawartość pamięci karty

Aby odczytać dowolny blok pamięci za pomocą czytnika ACR122U, postępuj zgodnie z poniższą procedurą:

1. ****Uruchom program:**** Otwórz plik `ACR122UTool.exe` z folderu aplikacji.
2. ****Nawiąż połączenie:**** Przejdź do zakładki ***Reader Commands*** i kliknij ****New connection****. Program połączy się z czytnikiem (na czytniku powinna zapalić się dioda). Połóż kartę na czytniku.
3. ****Pobierz ATR:**** W ***Reader Commands*** kliknij ****Get ATR**** (Answer To Reset). To polecenie sprawdza obecność karty, jej typ i często pozwala zidentyfikować jej podstawowe parametry (w tym UID, o czym wspomnę w samoweryfikacji).
4. ****Załaduj klucze do czytnika:**** Przejdź do ***Card Commands*** -> ****Load Authentication Keys****.
 - * ***Key number:*** `00` (wybieramy pierwszy slot pamięci czytnika na klucz).
 - * ***Key:*** `FF FF FF FF FF FF` (jest to domyślny, fabryczny klucz transportowy dla kart MIFARE).
5. ****Uwierzytelnienie (Autoryzacja):**** Zanim odczytasz blok, musisz udowodnić karcie, że masz do niego prawo. W ***Card Commands*** -> ****Authenticate****:
 - * ***Block No.:** wpisz numer bloku, który chcesz odczytać (np. `04` w notacji szesnastkowej).
 - * ***Key Type:*** `Type A` (standardowy typ klucza do odczytu w MIFARE).
 - * ***Key No.:** `0` (wskazujemy klucz załadowany w kroku 4).
6. ****Odczyt bloku:**** Po pomyślnej autoryzacji (program powinien zwrócić status `90 00` oznaczający sukces), przejdź do ***Card Commands*** -> ****Read Block****.
 - * ***Block numer:*** wpisz dokładnie ten sam numer bloku, który przed chwilą autoryzowałeś (np. `04`). Program wyświetli 16 bajtów danych zawartych w tym bloku.

2. Jak znaleźć tekst „LabPKE” w pamięci karty?

Program ACR122UTool odczytuje dane w formacie szesnastkowym (HEX). Aby znaleźć słowo „LabPKE”, musisz wiedzieć, jak wygląda ono w kodzie ASCII zapisanym szesnastkowo.

* Konwersja znaków na HEX:

* ****L**** = `4C`
* ****a**** = `61`
* ****b**** = `62`
* ****p**** = `50`
* ****K**** = `4B`
* ****E**** = `45`

* Szukany ciąg w pamięci karty to: ****`4C 61 62 50 4B 45`****.

* ****Proces szukania:**** Musisz autoryzować i odczytywać kolejne bloki pamięci (np. od `01`, pomijając `00`, który jest blokiem producenta) i wzrokowo szukać powyższego ciągu bajtów w oknie odpowiedzi programu.

3. Różnica między notacją szesnastkową (HEX) a dziesiętną (DEC)

* ****Dziesiętna (DEC):**** To system, którego używamy na co dzień (cyfry 0-9).

*****Szesnastkowa (HEX):**** To system używany w informatyce (cyfry 0-9 oraz litery A-F, gdzie A=10, B=11... F=15).

*****W kontekście ACR122UTool:**** Program ten domyślnie przyjmuje numery bloków w systemie *****HEX****. Jest to kluczowa różnica! Jeśli chcesz odczytać dziesiąty blok (dec: 10) i wpiszesz w programie "10", program zinterpretuje to jako blok 16 w systemie dziesiętnym (bo `10` hex = `16` dec). Aby odczytać dziesiąty blok, w programie musisz wpisać `0A`.

4. Jak przeliczyć numer bloku z notacji ciągłej na standard MIFARE?

Karta MIFARE Classic 1K posiada 64 bloki, numerowane w notacji ciągłej od 0 do 63. Są one podzielone na *****16 sektorów****, a każdy sektor ma po *****4 bloki**** (blok 0, 1, 2 na dane i blok 3 na klucze/bity dostępu).

Wzór na przeliczenie bloku ciągłego (w systemie dziesiętnym) na standard

Sektor/Blok:

*****Numer Sektora**** = Wynik z dzielenia całkowitego: `[Numer bloku DEC] / 4`

*****Numer Bloku w sektorze**** = Reszta z dzielenia: `[Numer bloku DEC] modulo 4`

5. Przykład przeliczenia dla bloku 9 (hex)

1. Krok 1: Przeliczamy numer bloku z HEX na DEC.

* `09` (hex) = *****9**** (dec).

2. Krok 2: Obliczamy Sektor.

* $9 / 4 = 2$ (z resztą 1).

*****Sektor = 2****

3. Krok 3: Obliczamy Blok w sektorze (reszta z dzielenia).

* $9 \text{ modulo } 4 = 1$.

*****Blok w sektorze = 1****

*****Wynik:**** Blok ciągły `09` (hex) to w standardzie MIFARE: *****Sektor 2, Blok 1****.

CZĘŚĆ 2: Samoweryfikacja rozwiązania

Teraz, zgodnie z instrukcją, przeprowadzam samoweryfikację na podstawie typowego raportu z odczytu (tworzę przykładowy, kompletny scenariusz odpowiedzi).

*****Lista kontrolna (Checklist):****

*****Czy UID jest podany w formacie 4 bajty szesnastkowo?*****

*****BŁĄD:**** W mojej pierwotnej analizie skupiłem się tylko na szukaniu tekstu, a instrukcja wymaga kompletnej odpowiedzi uwzględniającej UID.

*****POPRAWKA:**** Karta MIFARE Classic 1K posiada unikalny numer seryjny (UID) o długości 4 bajtów, np. *****A1 B2 C3 D4****. UID znajduje się zawsze w bloku ciągłym `00`. Został teraz poprawnie uwzględniony w wynikach.

*****Czy numer bloku jest podany w notacji szesnastkowej?*****

*****TAK:**** Podano blok `09` (hex).

*****Czy przeliczenie sektora i bloku jest zgodne ze wzorem?*****

*****TAK:**** Użyto dzielenia modulo 4. $9 / 4 = 2$, r. 1.

*****Czy odpowiedź jest kompletna (UID, blok hex, sektor dec, blok dec)?*****

*****TAK (po poprawce):**** Kompletne podsumowanie dla bloku `09`:

* UID karty: np. *****1A 2B 3C 4D**** (znajduje się w bloku 00)

* Blok ciągły (HEX): *****09****

* Sektor (DEC): *****2****

* Blok w sektorze (DEC): *****1****

CZĘŚĆ 3: Predykcja wyniku

Zadanie: Przewidź, jaki wynik otrzyma użytkownik dla bloku wpisanego w ACRTTool jako `0A` (hex).

****Obliczenia:****

1. ****Blok w ACRTTool (HEX):**** `0A`
2. ****Zamiana na DEC:**** `0A` w systemie szesnastkowym to ****`10`**** w systemie dziesiętnym.
3. ****Obliczenie Sektora:****
* $10 / 4 = 2$ (z resztą 2).
* ****Sektor = 2****
4. ****Obliczenie Bloku w sektorze:****
* $10 \bmod 4 = 2$.
* ****Blok = 2****

****Oczekiwany wynik dla użytkownika:****

Jeśli użytkownik wpisze w ACR122UTool do autoryzacji i odczytu blok ****`0A`****, w rzeczywistości odczyta ****Sektor 2, Blok 2**** w architekturze pamięci karty MIFARE Classic 1K.

Analiza poprawności rozwiązania

Kryterium	Punkty (0-5)
Kompletność	5
Czytelność	5
Poprawność techniczna	5
Samoweryfikacja	5
Trafność predykcji	5

Ogólna ocena modelu: 25/25

Odpowiedź modelu Gemini 3.1 Pro została oceniona wysoko, ponieważ była kompletna, czytelna i poprawna technicznie. Model poprawnie zastosował wzór przeliczenia bloku na sektor i blok w sektorze oraz trafnie przewidział wynik działania. Dodatkową zaletą była samoweryfikacja odpowiedzi, co pokazuje, że model potrafi kontrolować poprawność własnego rozwiązania.

2) Claude Sonet 4.5

Odpowiedź modelu
Kompleksowy przewodnik odczytu pamięci karty MIFARE
1. Krok po kroku: Odczyt zawartości pamięci karty MIFARE
Krok 1: Uruchomienie programu - Uruchom `ACR122U\Application\ACR122UTool.exe`

- Upewnij się, że czytnik ACR122U jest podłączony do komputera

Krok 2: Nawiązanie połączenia

- Przejdź do: ****Reader Commands → New connection****
- Program nawiąże komunikację z czytnikiem

Krok 3: Pobranie ATR (Answer To Reset)

- Przejdź do: ****Reader Commands → Get ATR****
- ****UID karty**** pojawi się w odpowiedzi (4 bajty w formacie hex, np. `3A 2B 1C 4D`)
- ATR zawiera podstawowe informacje o karcie

Krok 4: Załadowanie klucza autentykacji

- Przejdź do: ****Card Commands → Load Authentication Keys****
- Ustaw parametry:
 - ****Key number:**** `00`
 - ****Key:**** `FF FF FF FF FF FF` (domyślny klucz fabryczny)
- Klucz zostanie załadowany do pamięci czytnika

Krok 5: Autentykacja bloku

- Przejdź do: ****Card Commands → Authenticate****
- Ustaw parametry:
 - ****Block No.:**** `00` (lub inny blok do odczytu)
 - ****Key Type:**** `Type A`
 - ****Key No.:**** `0`
- Autentykacja jest wymagana przed każdym odczytem

Krok 6: Odczyt bloku

- Przejdź do: ****Card Commands → Read Block****
- Ustaw: ****Block number:**** `00` (lub inny blok)
- Dane zostaną wyświetlone w formacie hex (16 bajtów)

2. Jak znaleźć tekst „LabPKE” w pamięci karty

Metodyka wyszukiwania:

****A. Konwersja tekstu na hex:****

- `L` = `4C`
- `a` = `61`
- `b` = `62`
- `P` = `50`
- `K` = `4B`
- `E` = `45`
- ****Szukany ciąg:**** `4C 61 62 50 4B 45`

****B. Systematyczne przeszukiwanie:****

1. Zaczynaj od bloku `00` (hex)
2. Autentykuj i odczytaj kolejne bloki
3. ****UWAGA:**** Pomiń bloki sektorowe (co 4. blok: 03, 07, 0B, 0F...) - są to bloki konfiguracyjne
4. Szukaj sekwencji `4C 61 62 50 4B 45` w danych

****C. Optymalizacja wyszukiwania:****

- Karta MIFARE Classic 1K ma 16 sektorów (0-15)
- Każdy sektor ma 4 bloki
- Razem: 64 bloki (numeracja 0-63 dec, 00-3F hex)
- Sprawdzaj bloki: 00, 01, 02, 04, 05, 06, 08, 09, 0A... (pomijając co 4.)

****Przykład:**** Jeśli w bloku `09` (hex) zobaczysz:

...

4C 61 62 50 4B 45 20 31 32 33 00 00 00 00 00 00

...

To znaczy, że znalazłeś "LabPKE 123"

3. Różnica między notacją szesnastkową a dziesiętną

Notacja szesnastkowa (HEX):

- Podstawa: 16
- Cyfry: 0-9, A-F
- ****W ACR122UTool:**** numery bloków podawane są w HEX
- Przykład: `0A` (hex) = 10 (dec)

Notacja dziesiętna (DEC):

- Podstawa: 10
- Cyfry: 0-9
- ****W raportach/obliczeniach:**** często używamy dec
- Przykład: `10` (dec) = 0A (hex)

Tabela konwersji (0-15):

HEX	DEC	HEX	DEC
----	----	----	----
00	0	08	8
01	1	09	9
02	2	0A	10
03	3	0B	11
04	4	0C	12
05	5	0D	13
06	6	0E	14
07	7	0F	15

Konwersja:

- ****HEX → DEC:**** 2F (hex) = $2 \times 16 + 15 = 47$ (dec)
- ****DEC → HEX:**** 47 (dec) = $47 \div 16 = 2$ reszta 15 = 2F (hex)

4. Przeliczenie numeru bloku z notacji ciągłej na standard MIFARE

Standard MIFARE (sektor + blok w sektorze):

- ****Sektor:**** grupa 4 bloków
- ****Blok w sektorze:**** pozycja 0-3

Wzory przeliczeniowe:

...

Blok ciągły (dec) → Sektor i blok w sektorze:

Sektor (dec) = Blok_ciągły (dec) ÷ 4 (dzielenie całkowite)

Blok w sektorze (dec) = Blok_ciągły (dec) mod 4 (reszta z dzielenia)

...

Przykłady:

****Przykład 1:**** Blok ciągły = `10` (dec) = `0A` (hex)

- Sektor = $10 \div 4 = 2$ (dec)

- Blok w sektorze = $10 \bmod 4 = 2$ (dec)

- ****Odpowiedź:**** Sektor 2, blok 2

****Przykład 2:**** Blok ciągły = `7` (dec) = `07` (hex)

- Sektor = $7 \div 4 = 1$ (dec)

- Blok w sektorze = $7 \bmod 4 = 3$ (dec)

- ****Odpowiedź:**** Sektor 1, blok 3 (blok sektorowy - konfiguracja!)

****Przykład 3:**** Blok ciągły = `15` (dec) = `0F` (hex)

- Sektor = $15 \div 4 = 3$ (dec)

- Blok w sektorze = $15 \bmod 4 = 3$ (dec)

- ****Odpowiedź:**** Sektor 3, blok 3 (blok sektorowy)

5. Przykład przeliczenia dla bloku 9 (hex)

Dane wejściowe:

- Blok w ACR122UTool: `09` (hex)

Krok 1: Konwersja HEX → DEC

...

$09 \text{ (hex)} = 0 \times 16 + 9 = 9 \text{ (dec)}$

...

Krok 2: Obliczenie sektora

...

$\text{Sektor} = 9 \div 4 = 2 \text{ (reszta 1)}$

Sektor (dec) = 2

...

Krok 3: Obliczenie bloku w sektorze

...

$\text{Blok w sektorze} = 9 \bmod 4 = 1$

Blok w sektorze (dec) = 1

...

Wynik końcowy:

- ****Blok ciągły (hex):**** `09`
- ****Blok ciągły (dec):**** `9`
- ****Sektor (dec):**** `2`
- ****Blok w sektorze (dec):**** `1`
- ****Interpretacja:**** To jest drugi blok danych w sektorze 2

Wizualizacja:

...

Sektor 0: bloki 00, 01, 02, 03

Sektor 1: bloki 04, 05, 06, 07

Sektor 2: bloki 08, 09, 0A, 0B ← blok 09 tutaj

^ ^ ^ ^ ^ ^ ^ ^

blok 0 blok 1

...

SAMOWERYFIKACJA ROZWIĄZANIA

Sprawdzenie poprawności:

1. Format UID:

-  UID powinien być w formacie 4 bajtów hex
- Przykład poprawny: `3A 2B 1C 4D`
- Przykład niepoprawny: `3A2B1C4D` (brak spacji), `58` (za krótki)

2. Format numeru bloku:

-  W ACR122UTool: notacja HEX (`09`, `0A`, `1F`)
-  W obliczeniach: możliwe DEC, ale należy zaznaczyć
- Przykład poprawny: "Blok 09 (hex) = 9 (dec)"

3. Weryfikacja wzorów:

...

Sektor = Blok_dec ÷ 4

Blok_w_sektorze = Blok_dec mod 4

...

-  Dla bloku 9: $9 \div 4 = 2$, $9 \bmod 4 = 1 \rightarrow$ ****POPRAWNE****

4. Kompletność odpowiedzi:

Pełna odpowiedź powinna zawierać:

-  UID (jeśli odczytano): `XX XX XX XX` (hex)
-  Numer bloku (hex): `09`
-  Sektor (dec): `2`
-  Blok w sektorze (dec): `1`

Weryfikacja dla bloku 9:

...

Dane: Blok 09 (hex)

Krok 1: 09 (hex) $\rightarrow$ 9 (dec) ✓

Krok 2: $9 \div 4 = 2$ (sektor) ✓

Krok 3: $9 \bmod 4 = 1$ (blok w sektorze) ✓

Wynik: Sektor 2, blok 1 ✓

...

PREDYKCJA WYNIKU

Zadanie: Blok w ACRTool to `0A` (hex)

Obliczenia krok po kroku:

****Krok 1: Konwersja HEX → DEC****

...

$0A \text{ (hex)} = 0 \times 16 + 10 = 10 \text{ (dec)}$

...

****Krok 2: Obliczenie sektora****

...

$\text{Sektor} = 10 \div 4 = 2 \text{ (reszta 2)}$

$\text{Sektor (dec)} = 2$

...

****Krok 3: Obliczenie bloku w sektorze****

...

$\text{Blok w sektorze} = 10 \bmod 4 = 2$

$\text{Blok w sektorze (dec)} = 2$

...

🏠 WYNIK PREDYKCJI:

...

WYNIK DLA BLOKU 0A (hex)		
Blok ciągły (hex): 0A		
Blok ciągły (dec): 10		
Sektor (dec):	2	
Blok w sektorze (dec):	2	
INTERPRETACJA:		
Trzeci blok danych w sektorze 2		
(Blok danych - nie jest sektorowy)		

...

Wizualizacja w pamięci:

...

Sektor 2:

Blok 08 (hex) = 8 (dec) → Sektor 2, blok 0

Blok 09 (hex) = 9 (dec) → Sektor 2, blok 1

Blok 0A (hex) = 10 (dec) → Sektor 2, blok 2 ← TU JESTEŚMY
Blok 0B (hex) = 11 (dec) → Sektor 2, blok 3 (blok sektorowy)

...

Co użytkownik zobaczy w ACR122UTool:

Po wykonaniu "Read Block" dla bloku `0A`:

- Otrzyma 16 bajtów danych w formacie hex
- Przykład: `4C 61 62 50 4B 45 00 00 00 00 00 00 00 00 00 00`
- Jeśli to "LabPKE", zobaczysz te bajty na początku

PODSUMOWANIE NAJWAŻNIEJSZYCH INFORMACJI

🗝 Klucze do zapamiętania:

1. ****Domyślny klucz MIFARE:**** `FF FF FF FF FF FF`
2. ****Struktura karty:**** 16 sektorów × 4 bloki = 64 bloki
3. ****Bloki sektorowe:**** Co 4. blok (03, 07, 0B...) - NIE zapisuj danych!
4. ****Konwersja:**** Sektor = Blok÷4, Blok_w_sektorze = Blok mod 4
5. ****"LabPKE" w hex:**** `4C 61 62 50 4B 45`

⚠ Częste błędy:

- ❌ Pomylenie HEX z DEC przy wpisywaniu numeru bloku
- ❌ Próba zapisu do bloku sektorowego (3, 7, 11, 15...)
- ❌ Brak autentykacji przed odczytem
- ❌ Użycie niewłaściwego klucza

✅ Poprawny workflow:

1. New connection
2. Get ATR (zapisz UID!)
3. Load Key: `FF FF FF FF FF FF`
4. Authenticate blok (hex)
5. Read block (hex)
6. Powtórz 4-5 dla kolejnych bloków

****Powodzenia w odczycie karty MIFARE!**** 🎯

Analiza poprawności rozwiązania

Kryterium	Punkty (0-5)
Kompletność	5
Czytelność	5
Poprawność techniczna	5

Samoweryfikacja	5
Trafność predykcji	5

Ogólna ocena modelu: 25/25

Claude Sonnet 4.5 bardzo dobrze poradził sobie z zadaniem, odpowiedź była kompletna, czytelna i technicznie poprawna. Model dobrze sprawdził swój wynik, sam zauważył, czy wszystko się zgadza, i poprawnie przewidział rezultat. Najlepiej wypada też dlatego, że użył jasnych nagłówków, list i przykładów, więc łatwo było śledzić tok rozumowania.

3) ChatGPT 5.2

Odpowiedź modelu
1) Odczyt zawartości pamięci karty MIFARE w ACR122UTool.exe — krok po kroku **Uruchom narzędzie** - `ACR122U\Application\ACR122UTool.exe` **Połącz się z czytnikiem** - `Reader Commands → New connection` - Jeśli masz kilka czytników, wybierz właściwy ACR122U i połącz. **Sprawdź ATR (czy karta jest widoczna)** - Przyłóż kartę do czytnika. - `Reader Commands → Get ATR` - Otrzymasz ciąg bajtów ATR (to potwierdza komunikację karta–czytnik). **Załaduj klucz do uwierzytelniania (domyślny)** - `Card Commands → Load Authentication Keys` - **Key number:** `00` - **Key:** `FF FF FF FF FF FF` - To ładuje klucz do pamięci czytnika pod numerem 00. **Uwierzytelnij blok, który chcesz czytać** - `Card Commands → Authenticate` - **Block No.:** np. `00` (albo inny, który chcesz odczytać) - **Key Type:** `Type A` (Key A) - **Key No.:** `0` (czyli klucz załadowany jako 00) - Jeśli uwierzytelnienie się nie powiedzie, nie odczytasz danych z tego sektora/bloku. **Odczytaj blok (16 bajtów)** - `Card Commands → Read Block` - **Block number:** np. `00` (albo ten sam, który uwierzytelnieś) - Wynik będzie 16 bajtów danych (zwykle prezentowanych w hex). **Czytaj kolejne bloki** - Żeby przejrzeć pamięć, powtarzasz: - Authenticate (dla bloku w danym sektorze) - Read Block

- Typowo dla MIFARE Classic 1K: sektory mają po 4 bloki (0–3, 4–7, 8–11, ...), gdzie ****ostatni blok sektora**** to tzw. ****sector trailer**** (z kluczami i bitami dostępu).

2) Jak znaleźć tekst „LabPKE” w pamięci karty

1. ****Odczytuj bloki po kolei**** (np. od bloku `04`, `05`, `06`...).
2. W wynikach odczytu masz ****hex**** (np. `4C 61 62 50 4B 45 ...`).
3. Szukasz ASCII dla `LabPKE`:
 - `L` = `0x4C`
 - `a` = `0x61`
 - `b` = `0x62`
 - `P` = `0x50`
 - `K` = `0x4B`
 - `E` = `0x45`
4. Czyli w pamięci możesz wypatrywać sekwencji:
 - ****`4C 61 62 50 4B 45`****
5. Jeśli narzędzie pokazuje też interpretację ASCII (czasem bywa panel z tekstem), wtedy wystarczy „wzrokowo” zobaczyć `LabPKE`. Jeśli nie — porównujesz bajty hex z tabelą ASCII jak wyżej.

3) Różnica między notacją szesnastkową (hex) a dziesiętną (dec) przy numeracji bloków

- ****Hex (szesnastkowo)**** używa cyfr: `0–9` oraz `A–F`.
- Przykład: `0A` (hex) oznacza 10 (dec).
- ****Dec (dziesiętnie)**** używa cyfr: `0–9`.
- Przykład: `10` (dec).

W ACR122UTool pola typu ****Block No.**** często wpisuje się w ****hex**** (np. `0A`, `09`, `10`), ale interpretacja zależy od narzędzia/wersji — w Twoim zadaniu wprost zakładamy:

- `0A` (hex) = 10 (dec).

4) Przeliczenie numeru bloku z notacji ciągłej na „standard MIFARE” (sektor + blok w sektorze)

Dla ****MIFARE Classic 1K**** (najczęstszy przypadek w laboratoriach):

- Każdy sektor ma ****4 bloki**** (bloki 0–3 w sektorze).
- Numer bloku „ciągły” to numer absolutny: `0, 1, 2, 3, 4, 5, ...`

Wzory (dla sektorów 0–15 w 1K):

- ****sektor (dec)**** = `floor(block_abs_dec / 4)``
- ****blok_w_sektorze (dec)**** = `block_abs_dec % 4``

Uwaga praktyczna:

- ****blok_w_sektorze = 3**** oznacza trailer sektora (klucze + access bits), zwykle tego nie „szukasz” pod kątem tekstu.

5) Przykład przeliczenia dla bloku `9 (hex)`

1. Zamiana `9 (hex)` na dziesiętne:

- `0x09 = 9 (dec)`

2. Przeliczenie na sektor + blok w sektorze (MIFARE 1K, 4 bloki/sektor):

- sektor = $\text{floor}(9 / 4) = 2$

- blok w sektorze = $9 \% 4 = 1$

****Wynik:****

- ****blok (hex):**** `09`

- ****blok absolutny (dec):**** `9`

- ****sektor (dec):**** `2`

- ****blok w sektorze (dec):**** `1`

Samoweryfikacja rozwiązania (checklista)

1) Czy UID jest podany w formacie 4 bajty szesnastkowo?

- W tej odpowiedzi ****nie** podaję UID******, bo nie został dostarczony z odczytu (zrzutu) i nie da się go uczciwie „zgadnąć”.

- Poprawny format UID 4B (przykład): `DE AD BE EF` (hex, 4 bajty).

- Jeśli chcesz, wklej wynik z narzędzia (UID/ATR), a sprawdzę format.

2) Czy numer bloku jest podany w notacji szesnastkowej?

- Tak: używam `09`, `0A` jako przykłady ****hex****.

3) Czy przeliczenie sektora i bloku jest zgodne ze wzorem?

- Tak, dla MIFARE Classic 1K: dzielenie przez 4 i modulo 4.

4) Czy odpowiedź jest kompletna (UID, blok hex, sektor dec, blok dec)?

- Dla części przeliczeniowej: ****tak**** (blok hex, blok dec, sektor dec, blok w sektorze dec).

- Dla UID: ****nie****, bo brak danych wejściowych. To nie jest błąd merytoryczny przeliczeń, tylko brak informacji od użytkownika.

5) Czy widzę błąd?

- Nie widzę błędu w przeliczeniach.

- Jedyna uwaga: wzór „/4” dotyczy ****MIFARE Classic 1K**** (sektory 0–15). Dla 4K jest inaczej w sektorach > 31 (tam są sektory po 16 bloków). W Twoim zadaniu kontekst wskazuje na klasyczne 1K/labowe bloki, więc przyjęcie /4 jest zgodne.

Predykcja wyniku: blok `0A (hex) = 10 (dec)` → sektor i blok

Dane: - blok w ACRTool: **`0A` (hex)** - to jest **10 (dec)** Przeliczenie (MIFARE Classic 1K): - sektor = `floor(10 / 4) = 2` - blok w sektorze = `10 % 4 = 2` **Wynik predykcji:** - **blok (hex):**`0A` - **blok absolutny (dec):**`10` - **sektor (dec):**`2` - **blok w sektorze (dec):**`2`
--

Analiza poprawności rozwiązania

Kryterium	Punkty (0-5)
Kompletność	4
Czytelność	5
Poprawność techniczna	5
Samoweryfikacja	4
Trafność predykcji	5

Ogólna ocena modelu: 23/25

ChatGPT 5.2 poradził sobie również bardzo dobrze. Odpowiedź zawiera prawie wszystkie wymagane elementy: opis odczytu, wyszukiwanie tekstu, różnicę między hex i dec, i wzory przeliczeniowe. Brakuje jednak pełnego i jednoznacznego uzupełnienia wszystkich pól w końcowej ocenie, zwłaszcza krótkiej, zamkniętej odpowiedzi z ocenami punktowymi. Odpowiedź jest bardzo dobrze uporządkowana. Odpowiedź jest czytelna, logiczna i w dużej części kompletna, a obliczenia są poprawne. Samoweryfikacja jednak nie kończy się jednoznaczną oceną całego rozwiązania.

4) Grok 4.3

Odpowiedź modelu
1. Krok po kroku odczyt pamięci MIFARE w ACR122UTool.exe 1. Uruchom `ACR122UTool.exe` 2. **Reader Commands → New connection** 3. **Reader Commands → Get ATR** (sprawdzenie karty) 4. **Card Commands → Load Authentication Keys** - Key number: `00` - Key: `FF FF FF FF FF FF` 5. **Card Commands → Authenticate** - Block No.: `00` (lub inny blok) - Key Type: `Type A` - Key No.: `0` 6. **Card Commands → Read Block** - Block number: `00` (wprowadzany w formacie hex)

7. Odczytany blok pojawia się w oknie jako dane szesnastkowe + ASCII.

Powtarzasz kroki 5–6 dla kolejnych bloków.

****2. Znajdowanie tekstu „LabPKE”****

Po odczycie bloku przeglądasz sekcję ASCII (prawa strona okna). Szukasz ciągu znaków `LabPKE`. Jeśli tekst jest zapisany w danym bloku, pojawi się czytelnie w ASCII. Czasem tekst bywa rozbity na 2 bloki.

****3. Różnica między notacją szesnastkową a dziesiętną****

- ****Dziesiętna (dec)****: normalne liczenie 0–9, 10, 11...
- ****Szesnastkowa (hex)****: 0–9, A, B, C, D, E, F (w ACR122UTool blok podaje się właśnie w hex, np. `0A` = 10 dec).

Narzędzie ACR122UTool oczekuje numeru bloku w formacie hex.

****4. Przeliczanie numeru bloku (ciągłego) na strukturę MIFARE****

MIFARE Classic 1K ma 16 sektorów × 4 bloki.

Wzór:

- `sektor = block // 4`
- `blok\_w\_sektorze = block % 4`

****5. Przykład dla bloku 9 (hex)****

`09` hex = ****9**** dec
`sektor = 9 // 4 = 2`
`blok\_w\_sektorze = 9 % 4 = 1`
→ ****Sektor 2, blok 1****

****Predykcja wyniku (blok 0A hex)****

`0A` hex = ****10**** dec
`sektor = 10 // 4 = 2`
`blok\_w\_sektorze = 10 % 4 = 2`
→ ****Sektor 2, blok 2****

Rozwiązanie jest poprawne (brak błędów w przeliczeniach).

Analiza poprawności rozwiązania

Kryterium	Punkty (0-5)
Kompletność	3
Czytelność	4
Poprawność techniczna	5

Samoweryfikacja	2
Trafność predykcji	5

Ogólna ocena modelu: 19/25

Grok 4.3 poradził sobie dobrze w obliczeniach i zawiera kluczowe elementy zadania, ale odpowiedź jest mniej kompletna i mniej szczegółowa niż w przypadku innych modeli. Samoweryfikacja jest bardzo krótka i nie obejmuje wszystkich wymaganych kryteriów, co obniża ogólną jakość odpowiedzi. Odpowiedź jest czytelna i uporządkowana dzięki krótkim listom i nagłówkom, ale bardziej skondensowana i mniej szczegółowa niż w przypadku Claude czy ChatGPT. Poprawność techniczna jest bardzo dobra, krótka i zwięzła. Samoweryfikacja jest bardzo krótka i tylko ogólnie stwierdza, że rozwiązanie jest poprawne. Nie ma checklisty, analizy braków (np. UID), ani oceny kompletności, więc nie jest na poziomie najlepszych odpowiedzi.

5) Deepseek v4 flash

Odpowiedź modelu
Świetnie, przeanalizujemy to zadanie krok po kroku. Jako asystent ds. kart MIFARE i systemów kontroli dostępu, pomogę Ci wykonać wszystkie punkty. --- ## 1. Instrukcja odczytu zawartości pamięci krok po kroku (ACR122UTool.exe) Oto rozszerzona i skomentowana wersja instrukcji: **Uruchom program:** `ACR122U\Application\ACR122UTool.exe` *(Upewnij się, że czytnik jest podłączony i sterowniki są zainstalowane)* **Połącz się z czytnikiem:** `Reader Commands → New connection` *(Utworzenie połączenia USB z czytnikiem ACR122U)* **Pobierz ATR (Answer To Reset):** `Reader Commands → Get ATR` *(Odczytuje podstawowe informacje o karcie, w tym UID – zwykle 4 lub 7 bajtów)* **Załaduj klucz uwierzytelniający:** `Card Commands → Load Authentication Keys` `Key number → 00` `Key → FF FF FF FF FF FF` *(Jest to domyślny klucz fabryczny dla MIFARE Classic)* **Uwierzytelnij się do bloku:** `Card Commands → Authenticate` `Block No. → 0` (lub inny blok) `Key Type → Type A` `Key No. → 0` *(Proces ten "odblokowuje" dostęp do odczytu/zapisu w danym sektorze)*

6. ****Odczytaj blok:****

- `Card Commands → Read Block`
- `Block number → 00` (lub inny blok)
- *(Wynik pojawi się w oknie głównym jako 16 bajtów w hex)*

7. ****Znajdź tekst "LabPKE":****

- Odczytywane dane są w hex. Należy je przekonwertować na ASCII.
- Tekst "LabPKE" w hex to: `4C 61 62 50 4B 45`
- Szukaj tego ciągu w wynikach odczytu kolejnych bloków.

2. Jak znaleźć tekst "LabPKE" w pamięci karty

1. Odczytuj blok po bloku (np. od bloku 0 do 63 lub 127, w zależności od pojemności karty).
2. Dla każdego bloku konwertuj otrzymane 16 bajtów hex na tekst ASCII.
3. Sprawdź, czy w odczytanym tekście występuje ciąg "LabPKE".
4. Jeśli tak – zapisz numer bloku (hex) i jego zawartość.
5. Jeśli nie – przejdź do następnego bloku.

Uwaga: Nie wszystkie bloki są dostępne do odczytu (bloki 3, 7, 11 itd. to bloki kontrolne sektorów – zawierają klucze i prawa dostępu, ale w nich też może być tekst, jeśli został tam zapisany).

3. Różnica między notacją szesnastkową a dziesiętną (w kontekście numeracji bloków)

Cecha	Notacja szesnastkowa (hex)	Notacja dziesiętna (dec)
-----	-----	-----
Podstawa	16	10
Cyfry	0-9, A-F	0-9
Przykład	`0A` `10`	
Zapis w ACRTTool	`00`, `0A`, `3F`	nie używana bezpośrednio

****W narzędziu ACRTTool** numer bloku podaje się **w notacji szesnastkowej**.**
Np. jeśli chcesz odczytać blok dziesiętny 16, w ACRTTool wpisujesz `10` (hex).

4. Jak przeliczyć numer bloku z notacji ciągłej na standard MIFARE

****Standard MIFARE**** – bloki są grupowane w sektory:

- Każdy sektor zawiera 4 bloki (w MIFARE Classic 1K: 16 sektorów × 4 bloki = 64 bloki).
- Numeracja bloków jest ****ciągła**** od 0 do 63 (dla 1K) lub 0-127 (dla 4K).

****Wzór przeliczeniowy:****

```
sektor = floor(blok_ciągły / 4)
blok_w_sektorze = blok_ciągły % 4
---
```

Gdzie:

- `blok\_ciagly` – numer bloku w notacji ciągłej (np. 10)
- `sektor` – numer sektora (0-15 dla 1K)
- `blok\_w\_sektorze` – numer bloku w sektorze (0-3)

5. Przykład przeliczenia dla bloku 9 (hex)

1. ****Blok 9 (hex)**** = $9 \times 16^0 =$ ****9 (dec)****

2. Obliczamy:

- `sektor = floor(9 / 4) = floor(2,25) = 2`
- `blok\_w\_sektorze = 9 % 4 = 1`

3. Wynik:

- Numer ciągły (hex): `09`
- Numer ciągły (dec): `9`
- Sektor (dec): `2`
- Blok w sektorze (dec): `1`

Samoweryfikacja rozwiązania

Sprawdźmy poprawność:

Kryterium	Status	Uwagi
UID w formacie 4 bajty hex?	✓	W Get ATR pojawia się jako ciąg 8 znaków hex (np. `04 A1 B2 C3`)
Numer bloku w notacji hex?	✓	W ACRTTool wpisujemy np. `0A` zamiast `10`
Przeliczenie sektora/bloku zgodne ze wzorem?	✓	`floor(9/4)=2`, `9%4=1`
Odpowiedź kompletna?	✓	UID: `XXXX XXXX` (zależy od karty), blok hex: `09`, sektor dec: `2`, blok dec: `1`

****Wszystko poprawne.**** Gdybyś wpisał w ACRTTool `09` (hex), a myślał, że to dziesiętne 9 – to też jest poprawny blok ciągły. Przeliczenie jest identyczne.

Predykcja wyniku

Dane:

- Blok w ACRTTool = `0A` (hex) → dziesiętnie = 10

Obliczenia:

- `sektor = floor(10 / 4) = floor(2,5) = 2`
- `blok\_w\_sektorze = 10 % 4 = 2`

****Przewidywany wynik:****

Parametr Wartość
----- -----
Blok hex (ACRTool) `0A`
Blok dec 10
Sektor dec 2
Blok w sektorze dec 2

Podsumowanie
Masz teraz pełną instrukcję odczytu, sposób wyszukiwania tekstu, różnice między systemami liczbowymi oraz metodę przeliczania bloków. Jeśli potrzebujesz pomocy przy konkretnym odczycie (np. interpretacja danych lub analiza UID), daj znać!

Analiza poprawności rozwiązania

Kryterium	Punkty (0-5)
Kompletność	5
Czytelność	5
Poprawność techniczna	5
Samoweryfikacja	5
Trafność predykcji	5

Ogólna ocena modelu: 25/25

Deepseek v4 flash poradził sobie bardzo dobrze i odpowiedź jest bardzo zbliżona jakościowo do Claude Sonnet 4.5. Odpowiedź jest bardzo czytelna i dobrze uporządkowana dzięki nagłówkom, listom, tabeli konwersji oraz blokom kodu. Informacje są przedstawione w przejrzystej formie, co ułatwia śledzenie toku rozumowania. Samoweryfikacja jest dobrze wykonana i obejmuje kluczowe kryteria: format UID, format numeru bloku, zgodność wzorów i kompletność odpowiedzi.

6) Deepseek v4 pro Thinking

Odpowiedź modelu
Oto szczegółowe rozwiązanie zadania z odczytu pamięci karty MIFARE przy użyciu narzędzia ACR122UTool.exe.

1. Instrukcja odczytu pamięci krok po kroku
1. **Uruchom program** Otwórz plik `ACR122U\Application\ACR122UTool.exe`.
2. **Nawiąż połączenie z czytnikiem** `Reader Commands → New connection` – wybierz czytnik ACR122U z listy.
3. **Pobierz ATR (opcjonalnie)**

`Reader Commands → Get ATR` – pokaże odpowiedź karty (Answer To Reset), potwierdzając obecność karty.

4. ****Wczytaj klucze uwierzytelniające****

`Card Commands → Load Authentication Keys`

- ****Key number:**** `00`

- ****Key:**** `FF FF FF FF FF FF` (domyślny klucz fabryczny)

Kliknij `Load`.

5. ****Uwierzytelnij sektor****

Aby odczytać bloki z danego sektora, musisz najpierw uwierzytelnić się wobec jego bloku traila (trailer block).

`Card Commands → Authenticate`

- ****Block No.:**** wpisz numer bloku traila sektora (np. `03` dla sektora 0, `0B` dla sektora 2 itd.)

- ****Key Type:**** `Type A`

- ****Key No.:**** `0` (odpowiada kluczowi 00)

Po poprawnym uwierzytelnieniu stan karty zmieni się na ***Authenticated***.

6. ****Odczytaj blok****

`Card Commands → Read Block`

- ****Block numer:**** podaj adres bloku w formacie szesnastkowym (np. `00` dla pierwszego bloku w sektorze 0).

Wynik – 16 bajtów w hex – pojawi się w oknie logu.

****Uwaga:**** ACRTool przyjmuje numer bloku w trybie ciągłym (blok 0–63 dla MIFARE Classic 1K), a nie numer bloku wewnątrz sektora.

7. ****Przykład odczytu bloku 0****

- Authenticate: Block No. `03`, Key Type A, Key No. 0

- Read Block: `00`

Otrzymasz dane producenta, w tym unikatowy identyfikator karty (UID).

2. Jak znaleźć tekst „LabPKE” w pamięci

Ciąg ****„LabPKE”**** w kodowaniu ASCII ma długość 6 bajtów:

...

L → 0x4C

a → 0x61

b → 0x62

P → 0x50

K → 0x4B

E → 0x45

...

Szukamy więc sekwencji szesnastkowej ****`4C 61 62 50 4B 45`****.

****Procedura przeszukania:****

- Karta MIFARE Classic 1K ma 16 sektorów, każdy po 4 bloki. Dane użytkownika zapisuje się w blokach 0,1,2 każdego sektora (blok 3 to trailer).

- Aby przeszukać całą dostępną przestrzeń, wykonaj:

1. Uwierzytelnij się wobec sektora (np. dla sektora 0 – Block No. `03`, dla sektora 1 – `07`, sektor 2 – `0B`, ... ogólnie `sektor\*4 + 3`).
 2. Odczytaj kolejno bloki 0, 1, 2 tego sektora (`Read Block` z odpowiednim numerem ciągłym).
 3. Sprawdź, czy w odczytanych 16 bajtach występuje szukana sekwencja.
- Czynność powtórz dla wszystkich 16 sektorów. Gdy znajdziesz dopasowanie, odnotuj numer sektora i numer bloku.

Jeśli tekst został zapisany jednorazowo i mieści się w jednym bloku, znaleziona lokalizacja poda dokładny adres.

3. Różnica między notacją szesnastkową a dziesiętną w numeracji bloków

- ****Szesnastkowa (hex)**** – używana w interfejsie ACR122UTool. Adresy bloków wprowadzamy jako liczby w systemie szesnastkowym, np. `00`, `0A`, `3F`. Każda cyfra to 0–F.
- ****Dziesiętna (dec)**** – naturalny sposób liczenia w dokumentacji MIFARE. Ten sam blok co `0A` (hex) to `10` (dec).

Podczas pracy z narzędziem wszystkie wpisywane numery bloków (Authenticate, Read Block) są w hex. W obliczeniach sektorów łatwiej przechodzi się na dziesiętne: dzielenie przez 4 i resztę obliczamy na wartościach dziesiętnych.

4. Przeliczanie numeru bloku z notacji ciągłej na standard MIFARE (sektor, blok)

Dla karty MIFARE Classic 1K bloki są adresowane w sposób ciągły od 0x00 do 0x3F (0–63 dec). Każdy sektor zawiera dokładnie 4 bloki, więc:

- ****Numer sektora (dec):****
`sektor = continuous\_block // 4` (dzielenie całkowite)
- ****Numer bloku w sektorze (dec):****
`blok\_w\_sektorze = continuous\_block % 4` (reszta z dzielenia)

Wyniki podajemy zazwyczaj w dziesiętnej, bo sektory i bloki w obrębie sektora są numerowane od 0 w standardowej dokumentacji.

****Wzór:****

- continuous_block (hex) → wartość dziesiętna `B` →
`sektor = B div 4`
`blok = B mod 4`

5. Przykład przeliczenia dla bloku 9 (hex)

Blok `09` w hex ma wartość dziesiętną `9`.

- ****Sektor:**** `9 // 4 = 2`
- ****Blok w sektorze:**** `9 % 4 = 1`

Oznacza to, że ciągły blok `0x09` znajduje się w ****sektorze 2, bloku 1**** (drugi blok danych sektora 2).

Blok jest blokiem danych, ponieważ dla sektora 2 blok 3 (trailer) to adres ciągły `0x0B`.

****Przykład kompletnej odpowiedzi zgodnej z wymaganiami:****

Założmy, że odczytaliśmy blok 0 i otrzymaliśmy UID:

- ****UID (4 bajty hex):**** `04 5A 12 D7`
- ****Odczytany blok danych (hex):**** `09`
- ****Sektor (dec):**** `2`
- ****Blok w sektorze (dec):**** `1`

Komunikat końcowy:

UID: 04 5A 12 D7; blok ciągły: 0x09; sektor: 2; blok: 1.

Samoweryfikacja rozwiązania

1. ****UID w formacie 4 bajtów szesnastkowo**** – tak, podano `04 5A 12 D7`.
2. ****Numer bloku w notacji szesnastkowej**** – tak, `0x09`.
3. ****Przeliczenie sektora i bloku zgodne ze wzorem**** – tak: $9 // 4 = 2$, $9 \% 4 = 1$.
4. ****Kompletność odpowiedzi**** – zawiera UID, blok hex, sektor dec, blok dec – ****tak****.

Rozwiązanie jest poprawne i gotowe do wykorzystania.

Predykcja wyniku – blok 0A (hex) = 10 (dec)

Dla bloku ciągłego `0x0A` (10 dziesiętnie):

- ****Sektor:**** $10 // 4 = 2$
- ****Blok w sektorze:**** $10 \% 4 = 2$

Użytkownik otrzyma zatem:

UID: (przykładowy, np. 04 5A 12 D7); blok ciągły: 0x0A; sektor: 2; blok: 2.

Jest to trzeci blok sektora 2, który nadal jest blokiem danych (trailer sektora 2 to blok ciągły 0x0B).

Analiza poprawności rozwiązania

Kryterium	Punkty (0-5)
Kompletność	5
Czytelność	5
Poprawność techniczna	5
Samoweryfikacja	5
Trafność predykcji	5

Ogólna ocena modelu: 25/25

Samoweryfikacja jest dobrze wykonana i obejmuje wszystkie kluczowe kryteria: format UID, format numeru bloku, zgodność wzorów i kompletność odpowiedzi. Model jasno stwierdza, że rozwiązanie jest poprawne i gotowe do wykorzystania. Odpowiedź jest kompletna, bardzo czytelna, technicznie najbardziej precyzyjna a samoweryfikacja i predykcja są w pełni poprawne. To najlepsza odpowiedź pod względem szczegółowości i jakości technicznej. Samoweryfikacja jest dobrze wykonana i obejmuje wszystkie kluczowe kryteria: format UID, format numeru bloku, zgodność wzorów i kompletność odpowiedzi. Model jasno stwierdza, że rozwiązanie jest poprawne i gotowe do wykorzystania.

5. Podsumowanie

Wszystkie modele poradziły sobie dobrze i poprawnie rozwiązały zadania oraz wyjaśniły swój tok myślowy, ale odpowiedzi różniły się długością i szczegółowością.

Najlepszy pod względem technicznym był Deepseek v4 pro Thinking. Odpowiedź była najbardziej szczegółowa, zawierała poprawne wyjaśnienia, miał bardzo dobre samoweryfikację i obliczenia.

Najlepszy do prostych, codziennych zadań laboratoryjnych okazał się Deepseek v4 Flash. Odpowiedź była kompletna, czytelna i poprawna, ale jednocześnie szybka. Model ten jest najlepszym wyborem, gdy nie potrzebujemy nadmiernej szczegółowości i zależy nam na niskim zużyciu tokenów.

Claude Sonnet 4.5 też był bardzo dobry, odpowiedź była wzorowo uporządkowana i kompletna, ale dłuższa. Sprawdziłby się zdecydowanie lepiej do planowania wykonania bardziej skomplikowanych zadań.

Warto też wyróżnić Gemini pro 3.1, który uzyskał najwyższe oceny w każdej z ocenianych kategorii, również jest to model radzący sobie dobrze ze skomplikowanymi zadaniami i najczęściej wybierany jako subskrypcja AI w pracy.

ChatGPT 5.2 wypadł nieco słabiej ze względu na mniej dokładną samoweryfikację, co rzutowało na kompletność odpowiedzi, natomiast całość wypowiedzi jest również na wysokim poziomie.

Najgorzej wypadł Grok 4.3 z uwagi na zbyt bardzo krótką i mniej szczegółową odpowiedź, natomiast taki styl wypowiedzi ceniony jest przez użytkowników platformy X, na której jest szeroko wykorzystywany.

Podsumowując model rozumujący zgodnie z oczekiwaniami wykazał się najwyższą jakością, natomiast do prostych i codziennych zadań najlepszym wyborem pozostaje model flash.