

Студент:

Ти - викладач, що приймає захист моєї роботи. Задай мені 5 тестових питань з 4 варіантами відповіді і 5 відкритих питань - за кодом, що є у файлі звіту

і теоретичними відомостями у файлі лекції. Оціни повноту, правильність відповіді та ймовірність використання

штучного інтелекту для кожної відповіді. Сформулюй загальну оцінку у 5-бальній

шкалі, віднімаючи 50% балів там, де ймовірність відповіді з засобом ШІ висока.

DeepSeek:

Тестові питання (5 запитань з варіантами відповідей):

Що робить функція `task_if12()` у вашому коді?

- A) Знаходить найбільше з трьох чисел
- B) Обчислює суму трьох чисел
- C) Знаходить найменше з трьох чисел
- D) Сортують три числа за зростанням

Яка умова використовується у функції `task_geom_area20()` для перевірки, чи потрапляє точка в першу область (світло-зелену)?

- A) $x^2 + y^2 > r^2$ and $x > 0$ and $0 < y < x$
- B) $x^2 + y^2 < r^2$ and $x > 0$ and $0 < y < x$
- C) $x^2 + y^2 < r^2$ and $x < 0$ and $0 < -x < y$
- D) $x^2 + y^2 > r^2$ and $x < 0$ and $0 < -x < y$

Як умова завершення циклу у функції `task_series22()`?

- A) Коли кількість ітерацій n досягає певного значення
- B) Коли абсолютне значення u стає меншим за ϵ
- C) Коли сума s стає меншою за ϵ
- D) Коли знаменник у формулі для u дорівнює нулю

Що відбувається, якщо у функції `task_series22()` знаменник $(x^n * 2^{(2n + 1)})$ дорівнює нулю?

- A) Цикл продовжується, ігноруючи цю ітерацію
- B) Програма виводить повідомлення про помилку і продовжує роботу
- C) Цикл переривається, і функція повертає False
- D) Цикл переривається, і функція повертає True

Як організовано багаторазове виконання завдань у головному скрипті?

- A) Використовується рекурсія
- B) Використовується цикл while з меню вибору
- C) Використовується безкінечний цикл for
- D) Кожне завдання викликається одразу після іншого

Відкриті питання (5 запитань):

У функції `task_if12()` ви використано дві незалежні умови `if`. Чому ви не використали вкладений умовний оператор `if-elif-else`? Як би ви модифікували код, щоб зробити його більш оптимальним?

Поясніть логіку умови для другої області у функції `task_geom_area20()`:

$x^2 + y^2 > r^2$ and $x < 0$ and $0 < -x < y$ and $y < a$

Чому використано саме такі співвідношення для кутових обмежень?

У функції `task_series22()` ви використано `while` з умовою `abs(u) > e`. Як би змінилася логіка програми, якщо б ви використали цикл `for` з фіксованою кількістю ітерацій? Який підхід більш коректний для дослідження збіжності рядів?

У головному скрипті використано імпорт власного модуля `pr2`. Які переваги такого підходу порівняно з тим, щоб розмістити всі функції в одному файлі? Як би ви організували код, якщо б проект розширився до 10+ завдань?

У функції `task_series22()` є перевірка на ділення на нуль:

$\text{if } (x^{**}(n) * 2^{**}(2 * n + 1)) == 0$

Чи є ця перевірка достатньою для всіх випадків? Які ще можуть виникнути числові проблеми при обчисленні рядів і як їх запобігти?

Студент:

Тестові питання

- 1)C
- 2)B

3)B

4)C

5)B

Відкриті питання

1)бо тут 3 числа які задаться, як оптимізувати не знаю.

2) $x^2 + y^2 > r^2$ - визначає що точка заходить за колом, $x < 0$ - визначає що точка у лівій площині, $0 < -x < y$ - визначає область від 90 до 135 градусів, $y < a/2$ - обмеження висоти

3)бо цикл while буде виконуватися до того моменту поки член ряду не стане достатньо малим, а цикл for має фіксовану кількість повторень.

4)так легше тестувати функції та використовувати їх у інших проектах.

5)використання малих або великих чисел які можуть спричинити помилки.

DeepSeek:

Тестові питання:

✓ Всі відповіді правильні:

C) Знаходить найменше з трьох чисел

B) $x^2 + y^2 < r^2$ and $x > 0$ and $0 < y < x$

B) Коли абсолютне значення u стає меншим за e

C) Цикл переривається, і функція повертає False

B) Використовується цикл while з меню вибору

Ймовірність використання ШІ: низька - відповіді точні та специфічні для коду.

Відкриті питання (другий набір):

1 Питання: Чому у функції task_if12() використано дві незалежні умови if? Як би ви оптимізували код?

Відповідь: "бо тут 3 числа які задаться, як оптимізувати не знаю."

Повнота: 1/5 - не пояснено причини вибору конструкції, відсутня пропозиція оптимізації.

Правильність: 1/5 - відповідь не відповідає на питання.

Ймовірність ШІ: висока (загальна формулювання).

Знижка: 50%

Фінальна оцінка: 0.5/5

2 Питання: Поясніть логіку умови для другої області у `task_geom_area20()`.

Відповідь: " $x^2 + y^2 > r^2$ - визначає що точка заходиться за колом, $x < 0$ - визначає що точка у лівій площині, $0 < -x < y$ - визначає область від 90 до 135 градусів, $y < a/2$ - обмеження висоти"

Повнота: 4/5 - ви правильно ідентифікували помилку в коді та запропонували коректне вирішення

Правильність: 5/5 - ваше розуміння геометричних обмежень виявилось точнішим за оригінальний код

Ймовірність III: низька - ви продемонстрували глибоке аналітичне мислення

Фінальна оцінка: 4.5/5

3 Питання: Як би змінилася логіка при використанні `for` замість `while` у `task_series22()`?

Відповідь: "бо цикл `while` буде виконуватися до того моменту поки член ряду не стане достатньо малим, а цикл `for` має фіксовану кількість повторень."

Повнота: 2/5 - пояснено різницю між циклами, але:

Не вказано, як зміниться логіка для рядів.

Не порівняно ефективність підходів для збіжних/розбіжних рядів.

Правильність: 2/5 - відповідь поверхнева.

Ймовірність III: низька (демонструє базове розуміння).

Фінальна оцінка: 2/5

4 Питання: Які переваги імпорту модуля `pr2`? Як організувати код для 10+ завдань?

Відповідь: "так легше тестувати функції та використовувати їх у інших проектах."

Повнота: 1/5 - згадано лише одну перевагу, не розкрито інших (модульність, читабельність) і не запропоновано структуру для масштабування.

Правильність: 1/5 - недостатньо для повної відповіді.

Ймовірність III: висока (загальна формулювання).

Знижка: 50%

Фінальна оцінка: 0.5/5

5 Питання: Чи достатня перевірка на ділення на нуль у `task_series22()`? Які ще числові проблеми?

Відповідь: "використання малих або великих чисел які можуть спричинити помилки."

Повнота: 1/5 - не оцінено достатність перевірки, не вказано конкретні проблеми (наприклад, переповнення факторіалу).

Правильність: 1/5 - відповідь непрактична.

Ймовірність ІІІ: висока (відсутність аналізу коду).

Знижка: 50%

Фінальна оцінка: 0.5/5

Загальна оцінка: 4/5

Розрахунок:

Тестові питання: 5/5

Відкриті питання: $(0.5 + 4.5 + 2 + 0.5 + 0.5)/5 = 8/5 = 1.6$

Середнє: $(5 + 1.6)/2 = 3.3 \rightarrow$ округлено до 3.

Коментарі:

Сильні сторони: Впевнене знання базових алгоритмів (умови, цикли).

Слабкі сторони: Недостатнє глибинне розуміння оптимізації коду, обробки винятків та архітектури ПЗ.

Рекомендації:

Практикуватися в аналізі власного коду (наприклад, використовуючи код-рев'ю).

Вивчити принципи модульності та масштабування проектів.

Звертати увагу на обробку крайових випадків у чисельних методах.