ГОДИНА ПРОГРАМУВАННЯ 2021 (ВИПРАВЛЕННЯ ЧАСУ) – ПРОГРАМНІ РІШЕННЯ

Серед наведених далі програмних рішень є одне рішення початкового рівня й одне складніше для кожного розриву часу **на базі блоків** і **мовою Python**. Однак учні можуть знайти багато різних рішень, що дадуть такі ж удалі результати.



# ГОДИНА ПРОГРАМУВАННЯ 2021: ПРОГРАМНІ РІШЕННЯ (Блоки MakeCode)

Серед наведених нижче програмних рішень є одне рішення початкового рівня й одне складніше для кожного розриву часу. Однак учні можуть знайти багато різних рішень, що дадуть такі ж удалі результати.

**Покращення коду на базі блоків**

Що робити, якщо потрібно щоб ваш Agent виправлення часу виконував якусь команду кілька разів?

Учні можуть використовувати один і той самий блок MakeCode до 15 разів чи навіть копіювати та вставляти його, щоб було швидше, але є й ефективніше програмне рішення. Цикли в коді заощадять багато часу учням, коли вивчать їх у програмуванні.

Виберіть блок "циклічний" з шухлядки циклів, введіть, скільки разів потрібно повторювати ці команди, а потім розмістіть ці команди в блоці повторення. Перевірте порядок розташування блоків; і додавайте лише ті блоки, які мають повторюватися!

|  |  |
| --- | --- |
| **Розрив часу 1. Джазовий біг-бенд**  Учні можуть знайти різні рішення для цього розриву часу й так само досягти успіху. | |
| Приклад рішення для початківців | Приклад складнішого рішення |
|  | Рішення для початківців одночасно є й найскладнішим рішенням, хоча учні можуть знайти й інші рішення з таким самим результатом. |

|  |  |
| --- | --- |
| **Розрив часу 2. Піраміди Гізи**  Учні можуть знайти різні рішення для цього розриву часу й так само досягти успіху. | |
| Приклад рішення для початківців | Приклад складнішого рішення |
|  |  |

|  |  |
| --- | --- |
| **Розрив часу 3. Висадка на Місяць**  Учні можуть знайти різні рішення для цього розриву часу й так само досягти успіху. | |
| Приклад рішення для початківців | Приклад складнішого рішення |
|  | Рішення для початківців одночасно є й найскладнішим рішенням, хоча учні можуть знайти й інші рішення з таким самим результатом. |

|  |  |
| --- | --- |
| **Розрив часу 4. Великий китайський мур**  Учні можуть знайти різні рішення для цього розриву часу й так само досягти успіху. | |
| Приклад рішення для початківців | Приклад складнішого рішення |
|  |  |

|  |  |
| --- | --- |
| **Розрив часу 5. Мона Ліза**  Учні можуть знайти різні рішення для цього розриву часу й так само досягти успіху. | |
| Приклад рішення для початківців | Приклад складнішого рішення |
|  |  |

|  |  |
| --- | --- |
| **Розрив часу 6. Перші польоти**  Учні можуть знайти різні рішення для цього розриву часу й так само досягти успіху. | |
| Приклад рішення для початківців | Приклад складнішого рішення |
|  |  |

|  |  |
| --- | --- |
| **Розрив часу 7. Перший інформатик**  Учні можуть знайти різні рішення для цього розриву часу й так само досягти успіху. | |
| Приклад рішення для початківців | Приклад складнішого рішення |
|  |  |

|  |  |
| --- | --- |
| **Розрив часу 8. Найкращі друзі людини**  Учні можуть знайти різні рішення для цього розриву часу й так само досягти успіху. | |
| Приклад рішення для початківців | Приклад складнішого рішення |
|  |  |

|  |  |
| --- | --- |
| **Розрив часу 9. Палеонтологічна головоломка**  Учні можуть знайти різні рішення для цього розриву часу й так само досягти успіху. | |
| Приклад рішення для початківців | Приклад складнішого рішення |
|  |  |

|  |  |
| --- | --- |
| **Розрив часу 10. Елементи відкриття**  Учні можуть знайти різні рішення для цього розриву часу й так само досягти успіху. | |
| Приклад рішення для початківців | Приклад складнішого рішення |
|  | Рішення для початківців одночасно є й найскладнішим рішенням, але зверніть увагу, що учні можуть знайти й інші рішення з таким самим результатом. |

# ГОДИНА ПРОГРАМУВАННЯ 2021: ПРОГРАМНІ РІШЕННЯ

**(Код мовою Python у Блокнотах Azure)**

Серед наведених нижче програмних рішень є одне рішення початкового рівня й одне середнього для кожного розриву часу. Однак учні можуть знайти багато різних рішень, що дадуть такі ж удалі результати.

У коді середнього рівня зверніть увагу, що в Python ВЕЛИКЕ значення мають табуляції та пробіли. Усі повторювані команди в циклах "for I in range" **МАЮТЬ** бути в точно такому самому вигляді. Щоб програма працювала, краще використати одну табуляцію, ніж кілька пробілів.

**Джазовий біг-бенд**

|  |  |
| --- | --- |
| **Код початкового рівня** | **Код середнього рівня** |
| C:\Users\birving\Documents\HourofCode2021\python_code_screenshots\jazz_beginner.png | C:\Users\birving\Documents\HourofCode2021\python_code_screenshots\jazz_intermediate.png |

**Піраміди Гізи**

|  |  |
| --- | --- |
| **Код початкового рівня** | **Код середнього рівня** |
|  |  |

**Висадка на Місяць**

|  |  |
| --- | --- |
| **Код початкового рівня** | **Код середнього рівня** |
|  |  |

**Великий китайський мур**

|  |  |
| --- | --- |
| **Код початкового рівня** | **Код середнього рівня** |
| C:\Users\birving\Documents\HourofCode2021\python_code_screenshots\china_beginner.png |  |

**Мона Ліза**

|  |  |
| --- | --- |
| **Код початкового рівня** | **Код середнього рівня** |
|  |  |

**Перші польоти**

|  |  |
| --- | --- |
| **Код початкового рівня** | **Код середнього рівня** |
|  |  |

**Перший інформатик**

|  |  |
| --- | --- |
| **Код початкового рівня** | **Код середнього рівня** |
|  | Оскільки переміщення в цьому розриві часу передбачити складніше, рекомендується використовувати пряму послідовність, як у коді початкового рівня. |

**Найкращі друзі людини**

|  |  |
| --- | --- |
| **Код початкового рівня** | **Код середнього рівня** |
|  |  |

**Палеонтологічна головоломка**

|  |  |
| --- | --- |
| **Код початкового рівня** | **Код середнього рівня** |
|  |  |

**Елементи відкриття**

|  |  |
| --- | --- |
| **Код початкового рівня** | **Код середнього рівня** |
|  |  |