Міністерство освіти і науки України

Департамент освіти освіти і науки Сумської облдержадміністрації

Сумське територіальне відділення МАН України

Відділення: комп’ютерних наук

Секція: Internet-технології

та WEB дизайн

ВЕБ-ОРІЄНТОВАНИЙ ДОДАТОК   
«ЗАХОПЛЮЮЧА ТЕОРІЯ ЧИСЕЛ»

Роботу виконав:

Кролевецький Денис Юрійович,

учень 10 класу Сумської спеціалізованої школи І-ІІІ ступенів №10 ім. Героя Радянського Союзу О.Бутка, м. Суми, Сумської області

Науковий керівник:

Фільченко Дмитро Вікторович

к. ф.-м. н., доцент кафедри моделювання складних систем Сумського державного університету

Суми — 2015

ВЕБ-ОРІЄНТОВАНИЙ ДОДАТОК «ЗАХОПЛЮЮЧА ТЕОРІЯ ЧИСЕЛ»

Кролевецький Денис Юрійович

комунальна установа Сумська спеціалізована школа

І-ІІІ ступенів № 10 імені Героя Радянського Союзу О. Бутка,

клас: 10-ФМ, м. Суми

Фільченко Дмитро Вікторович,

к. ф.-м. н., доцент кафедри моделювання складних систем Сумського державного університету

Метаданої роботи — розробка оптимального (за кількістю операцій) алгоритму переходу від одного заданого числа до іншого заданого числа з використанням операцій віднімання, множення, ділення на число 3, та програмна реалізації цього алгоритму у вигляді веб-орієнтованого додатку.

Актуальністьроботи полягає у можливості застосування результатів у багатьох сферах, таких як криптографія та обчислювальні технології, а також у перспективі використання результатів у подальших фундаментальних та прикладних наукових дослідженнях. Крім цього, розроблений веб-додаток дає змогу школярам розібратися у цікавих темах, які не входять до шкільної програми, і заохочує до вивчення математики та до знайомства з теорією чисел.

В роботі поставлені наступні завдання: розробити алгоритм перетворення заданого числа *m* в задане число *n* за мінімальну кількість операцій віднімання, множення, ділення з використанням числа 3; програмно реалізувати алгоритм у вигляді веб-орієнтованого додатку; передбачити можливість виведення формул, проміжних результатів та коментарів.

Результатом роботи є алгоритм, який програмно реалізований у вигляді html-сторінки. Написано скрипт, який реалізує роботу розробленого алгоритму. Крім цього, за допомогою скрипту передбачено валідацію вхідних даних та виведення коментарів, формул та проміжних результатів. Такий додаток може використовуватися школярами для знайомства з можливостями теорії чисел, а розроблений алгоритм може бути покладений в основу подальших досліджень.

ЗМІСТ

[ВСТУП 4](#_Toc413017521)

[РОЗДІЛ 1 СУЧАСНІ ВЕБ-ДОДАТКИ – ЇХ РОЛЬ ТА ЗАСТОСУВАННЯ 7](#_Toc413017522)

[1.1. Принципи роботи та особливості веб-додатків 7](#_Toc413017523)

[1.2. Місце та роль веб-додатків у навчанні 8](#_Toc413017524)

[1.3. Приклади веб-додатків, що застосовуються у навчальному процесі 9](#_Toc413017525)

[РОЗДІЛ 2 ВИБІР ЗАСОБІВ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ 13](#_Toc413017526)

[2.1 Вибір математичного апарату 13](#_Toc413017527)

[2.2 Вибір комп’ютерних засобів для програмної реалізації 14](#_Toc413017528)

[РОЗДІЛ 3 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ 16](#_Toc413017529)

[3.1 Побудова алгоритму розрахунків 16](#_Toc413017530)

[3.2 Схема роботи та структура додатку 23](#_Toc413017531)

[3.3 Тестування роботи додатку та аналіз результатів 26](#_Toc413017532)

[ВИСНОВКИ 30](#_Toc413017533)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 31](#_Toc413017534)

[ДОДАТОК А 32](#_Toc413017535)

[ДОДАТОК Б 33](#_Toc413017536)

[ДОДАТОК В 35](#_Toc413017537)

## ВСТУП

Кожну людину весь час оточують числа — чи то у повсякденному житті, чи на роботі, чи навіть на відпочинку. Починаючи з дитинства ми вчимося виконувати дії над ними. Спочатку найпростіші дії, а потім далі й ділі відкриваємо для себе захоплюючий світ чисел. Математика лежить в основі всіх наук і зустрічається у всіх галузях, тому її вивченню приділяється значна увага.

Однією з великих галузей математики є теорія чисел [2]. Інколи її називають вищою арифметикою або наукою про цілі числа. Вона присвячена вивченню властивостей натуральних чисел, тобто таких, що виникають природним чином при лічбі. Це числа: 1, 2, 3, 4, … Множину натуральних чисел прийнято позначати знаком N.

Всі натуральні числа розподіляються на 2 класи: до 1-го класу відносяться числа, що мають два натуральні дільники, а саме одиницю і самого себе, до 2-го — всі інші. Числа 1-го класу стали називати простими, а 2-го — складеними.

Властивості простих чисел і їх зв'язок зі всіма натуральними числами вивчалися Евклідом (3 ст. до н.е.). І дотепер властивості чисел постійно вивчаються. Шкільна програма теж не оминає це питання. Цікаві задачі теорії чисел часто виносяться на математичні турніри та олімпіади. Наприклад, деякі к таких задач приведені в джерелі [4]. Так, у 2013 році одна з задач ХVІ Всеукраїнського турніру юних математиків імені професора М.Й. Ядренка полягала в тому, що необхідно було знайти мінімальну кількість операцій, за які число *m* можна перетворити в число *n*, використовуючи лише операції віднімання, ділення, множення на число 3. Отже, постало питання — яким чином визначити мінімальну кількість операцій для переходу з одного числа в інше, та як визначити послідовність цих операцій. Задача ця цікава як з точки зору можливого застосування її результатів, так і навіть для простих школярів, адже демонструє властивості чисел, чим викликає зацікавлення та заохочує їх до математики.

Оптимальний (за кількістю операцій) алгоритм перетворення чисел може знайти застосування наприклад у криптографії. Ця наука постійно розвивається і зараз, як ніколи раніше, потрібні нові і надійні способи захисту інформації. Тому, такий алгоритм може стати частиною криптографічного методу.

З бурхливим розвитком трійкових комп’ютерів нове життя отримала трійкова система числення — позиційна система числення з цілочисленною основою, рівною 3. Спосіб перетворення чисел в використанням операцій з числом 3 може знайти застосування у цьому новому науковому напрямку. Тобто, ідея створення вищеописаного алгоритму є досить перспективною з точки зору можливих застосувань.

Для розуміння принципів роботи алгоритму можна програмно реалізувати його у вигляді додатку. Краще всього, якщо додаток буде веб-орієнтованим, адже з ним зможе познайомитись широка аудиторія користувачів. Крім цього, автоматизація обчислень забезпечить більш точний результат і захистить від випадкових помилок. Таким чином, додаток не лише допоможе розібратися у роботі алгоритму, а й автоматизувати обчислення для отримання точного результату.

Отже, тематика даної роботи є **актуальною**, а її вибір — **доцільним,** з огляду на можливість застосування результатів роботи у багатьох сферах, таких як криптографія та обчислювальні технології, а також з огляду на фундаментальність дослідження і перспективу використання результатів у подальших фундаментальних та прикладних дослідженнях. Крім цього тематика роботи є близькою автору як представнику школярів, адже дає змогу розібратися у цікавих темах, які не входять до шкільної програми, і заохочує до вивчення математики та до знайомства з теорією чисел.

**Мета** даної роботи полягає у розробці алгоритму оптимального з точки зору кількості операцій переходу від одного заданого числа до іншого заданого числа з використанням операцій віднімання, множення, ділення на число 3, а також у програмній реалізації розробленого алгоритму у вигляді веб-орієнтованого додатку.

**Об’єктом** дослідження є оптимальний алгоритм перетворення числа в число, а **предметом** дослідження є програмна реалізація розробленого алгоритму у вигляді веб-орієнтованого додатку «Захоплююча теорія чисел».

З огляду на вищенаведене, зміст поставлених в роботі **завдань** полягає в наступному: розробити алгоритм перетворення заданого числа *m* в задане число *n* за мінімальну кількість операцій віднімання, множення, ділення з використанням числа 3; програмно реалізувати алгоритм у вигляді веб-орієнтованого додатку; передбачити можливість виведення проміжних результатів та пояснень; передбачити валідацію даних, які вводяться користувачем.

Основними **методами** дослідження є: метод математичного аналізу; розробка веб-сторінок на мові HTML з використанням каскадних таблиць стилів (CSS); основи програмування на мові JavaScript.

Робота має як **теоретичну цінність** (побудований алгоритм може бути покладений в основу фундаментальних досліджень), так і **практичну цінність** (розроблений додаток може бути використаний у навчальному процесі для вивчення роботи алгоритму та для заохочення школярів до поглибленого вивчення математики; розроблений алгоритм може бути використаний як основа для подальших розробок в таких напрямках, як криптографія, трійкова система числення та ін.).

**Наукова новизна** роботи полягає в розробці нового алгоритму та програмній реалізації цього алгоритму.

Ідея роботи та допомога в проведенні літературного огляду належать науковому керівнику роботи. Розробка алгоритму проводилась спільно   
з к.ф.-м. н., доцентом кафедри моделювання складних систем Назаренком О.М. Виконання ж усіх інших завдань становить **особистий внесок** учасника конкурсу-захисту науково-дослідних робіт.

## РОЗДІЛ 1 СУЧАСНІ ВЕБ-ДОДАТКИ ­– ЇХ РОЛЬ ТА ЗАСТОСУВАННЯ

### Принципи роботи та особливості веб-додатків

Щорічно, щоденно, щогодинно кількість користувачів Інтернет стрімко зростає. Люди вже не уявляють свого повсякденного життя без доступу до всесвітньої мережі. Для цього вони використовують різноманітні пристрої — персональні комп’ютери, мобільні телефони, планшети, телевізори. Але мета всіх користувачів одна — це інформація. Вона може буті різного формату, тематики, однак сьогодні цінність будь-якої інформації вимірюється швидкістю доступу до неї. Динамічна сучасність накладає вимоги до інформації, повнота та якість стає другорядною, а швидкість та зручність доступу виходить на перший план. Тому, більшість сервісів переміщується у веб-простір. Так, веб-орінтовані додатки замінюють десктопні додатки (Windows додатки), адже вони мають ряд переваг [5].

Розглянемо особливості веб-додатків більш детально. Отже, веб-додаток [6] — це розподілений додаток, в якому клієнтом виступає браузер, а сервером — веб-сервер. Браузер здатний відображати веб-сторінки і, зазвичай, входить до складу операційної системи, а його оновлення і супровід здійснює постачальник операційної системи.

Логіка веб-додатку, як правило, зосереджується на сервері, а функція браузера полягає переважно у відображенні інформації, яка завантажується з сервера, і передачі назад даних користувача. Однією з переваг такого підходу є той факт, що клієнти не залежать від конкретної операційної системи користувача, тому веб-додатки є міжплатформовими сервісами.

В залежності від того, де відбувається інтерпретація вмісту веб-сторінок (на стороні клієнта чи сервера), всі веб-технології можна розділити на дві категорії: клієнтські та серверні [1]. Перші — інтерпретуються браузером. До них належать такі технології, як HTML, CSS, JavaScript, AJAX, XML, Flash, Silverlight та ін. [7]. До другої категорії відносяться ті технології, які інтерпретується на стороні сервера. В такому випадку на стороні сервера за допомогою певного програмного забезпечення, а саме веб-сервера, відбувається перетворення у формат, зрозумілий клієнтській стороні. Серверним є такі технології, як PHP, SQL, VBScript та ін.

Істотною перевагою побудови додатків у веб-орієнтованому форматі є те, що робота додатку має відбуватися незалежно від операційної системи клієнта. Замість того, щоб писати різні версії для Microsoft Windows, Mac OS X, GNU/Linux й інших операційних систем, додаток створюється один раз для довільно обраної платформи і на ній розгортається. Проте різна реалізація HTML, CSS, DOM й інших специфікацій в браузерах може викликати проблеми при розробці веб-додатків і подальшій їх підтримці. Крім того, можливість користувача налаштовувати багато параметрів браузера (наприклад, розмір шрифту, кольори, відключення підтримки сценаріїв) може перешкоджати коректній роботі додатку.

### Місце та роль веб-додатків у навчанні

Стрімкий розвиток технологій зробив комп’ютерну техніку невід’ємною частиною навчального процесу. Більш того, нові завдання сучасного суспільства активізують нові методи організації навчального процесу. З огляду на це, важливого значення набуває питання використання у педагогічному процесі мультимедійних засобів навчання (МЗН). Під МЗН розуміємо сукупність різних засобів навчання, таких як текстів, графічних зображень, музики та відео, додатків, спрямованих на здобуття знань, вдосконалення вмінь та навичок учнів. Вони забезпечують потрібну наочність у навчанні, а також можливість багаторазового відтворення спеціально переробленої навчальної інформації, яка відбиває сутність об’єктів, процесів і явищ, що вивчаються.

Запровадження в навчальний процес сучасних інформаційних, зокрема, комп’ютерних і телекомунікаційних технологій, відкриває нові шляхи й надає широкі можливості для подальшої диференціації загального навчання, всебічної активізації творчих, пошукових, особистісно-орієнтованих комунікативних форм навчання, підвищення його ефективності, мобільності й відповідності запитам сучасного світу.

Використання мультимедійних засобів у навчальному процесі з метою повторення, узагальнення та систематизації знань не тільки допомагає створити конкретне, наочно-образне уявлення про предмет, явище чи подію, які вивчаються, але й доповнити відоме новими даними. Відбувається не лише процес пізнання, відтворення та уточнення вже відомого, але й поглиблення знань.

Отже, застосування навчальних програм надає вивченню нового матеріалу специфічну новизну, яка за своїм змістом і формою викладення має можливість відтворити за короткий час значний за обсягом матеріал, а також подати його в незвичному аспекті, викликати в школярів нові образи, деталізувати нечітко сформовані уявлення, поглибити здобуті знання. Крім цього, таким чином, можна викликати інтерес до тематик, які раніше не викликали особливого зацікавлення у учнів і виглядали складними, непосильними.

Однією з перепон на шляху впровадження навчальних додатків у навчальний процес є доступ до них. Адже, найкращий ефект досягається, коли учень має змогу працювати з додатком і на уроці в присутності вчителя, й вдома. Але не завжди є змога працювати в ком’ютерному класі, а вдома у учня можуть виникати технічні проблеми, які він самостійно не зможе подолати. Вирішенням таких проблем є розміщення навчальних матеріалів у веб-просторі, а саме організація додатків у веб-орінтованому форматі.

### Приклади веб-додатків, що застосовуються у навчальному процесі

Одним з найвідоміших в Україні сервісів, який був інтегрований в навчальний процес, став проект «Щоденник.ua» <http://shodennik.ua/> (див. рис.1.1).

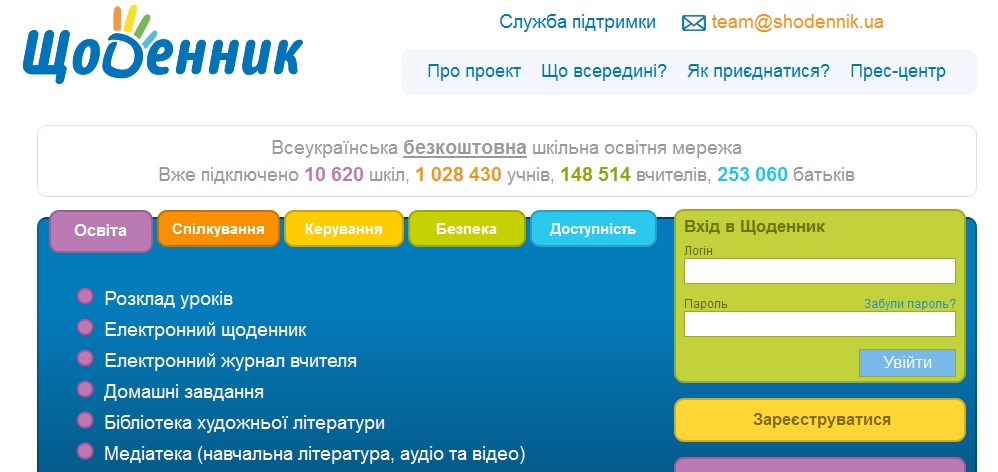


Рис.1.1 — Головна сторінка ресурсу «Щоденник.ua»

Як повідомляє сам веб-сайт, «Щоденник.ua» — це всеукраїнська безкоштовна освітня мережа, яка формує унікальне електронне середовище для вчителів, учнів та їх батьків. Метою проекту є об'єднання всіх педагогів, учнів та батьків України в єдину спільноту, модернізація навчального процесу та впровадження сучасних комп'ютерних технологій у школах. Ресурс надає різноманітні функції для школярів, учителів і батьків, такі як розклад уроків, електронний журнал, домашні завдання, словники та перекладачі, файли та інше. Таким чином, з будь-якого місця і пристрою вчителя, учні чи їх батьки, зможуть отримати доступ до всього необхідного.

Всесвітню популярність отримав ресурс «Wolfram|Alpha» <http://www.wolframalpha.com> (див. рис. 1.2). Це — база знань і набір обчислювальних алгоритмів.

«Wolfram|Alpha» не видає перелік посилань, що ґрунтується на результатах запиту, а обчислює відповідь, ґрунтуючись на власній базі знань, яка містить дані з математики, фізики, астрономії, хімії, біології, медицини, історії, географії, політики, музики, кінематографії, а також інформацію про відомих людей та інтернет-сайти. Він здатний переводити дані між різними одиницями вимірювання, системами числення, підбирати загальну формулу послідовності, обчислювати суми, границі, інтеграли, розв'язувати рівняння і системи рівнянь, проводити операції з матрицями, визначати властивості чисел та багато іншого.

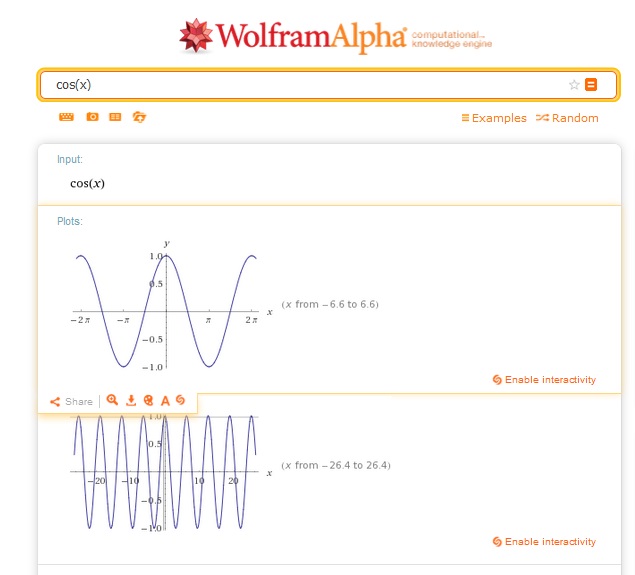


Рис.1.2 — Приклад пошуку інформації в «Wolfram|Alpha»

Також відомим є ресурс «MathPapa» <http://mathpapa.com> (див. рис. 1.3). Це — зручний та функціональний обчислювальний сервіс математичних операцій, за допомогою якого можна вивчати математику в інтерактивному режимі. Ресурс пропонує багато різних цікавих завдань, які дозволяють перевірити рівень математичних здібностей, швидкість розв’язування і якість отриманих знань.

«MathPapa» може розв’язувати заданий приклад, вміє будувати графіки різних функцій у вказаному діапазоні, та схематично показує значення дробів за допомогою кругових діаграм. Особливістю даного сервісу є те, що при розв’язанні відображаються всі кроки. Тому цей ресурс можна використовувати як для обчислень, так і для навчання. Але розробники все ж попереджають, що не можна на 100% довіряти ресурсу, адже він не досконалий і використовувати його слід на власний ризик.

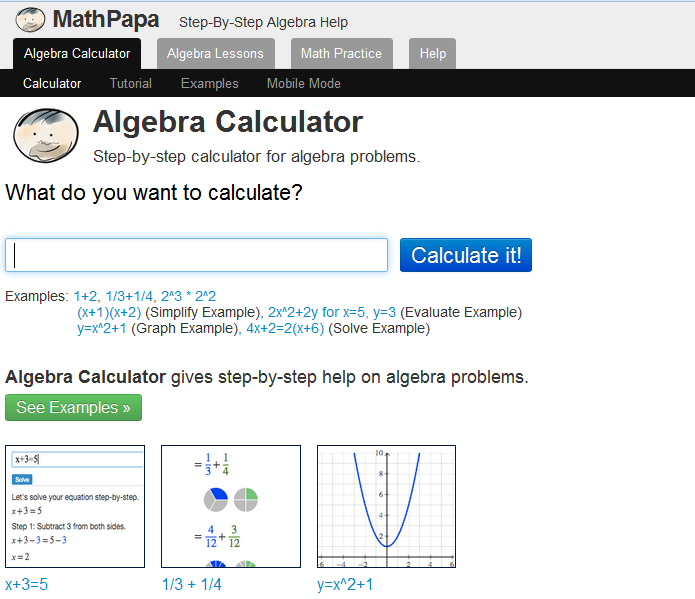


Рис.1.3 — Головна сторінка «MathPapa»

Отже, такі ресурси, як розглянуті вище, значно підвищують ефективність навчального процесу. Також слід зазначити, що веб-орінтовані додатки мають ряд переваг, за рахунок того, що не потребуються встановлення на конкретний пристрій та доступ до них можна отримати з будь-якого приладу, який має браузер та доступ до всесвітньої мережі.

## РОЗДІЛ 2 ВИБІР ЗАСОБІВ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

### Вибір математичного апарату

Розглядаючи поставлену задачу, а саме побудову алгоритму пошуку мінімальної послідовності операцій віднімання, ділення чи множення з використанням числа 3, за допомогою якої можна отримати задане число *n* з заданого числа *m*, слід почати з дослідження властивостей натуральних чисел [4] та розгляду більш простої ситуації.

Отже, нехай *n* — деяке натуральне число, *n*+1 — наступне за ним.

Продемонструємо можливість переходу із наступного числа в попереднє. Не важко помітити, що:

. (2.1)

Тобто перехід завжди можна здійснити за три кроки. А оптимальність кроків обґрунтовується неможливістю переходу за два або один крок.

Отже, перший висновок: число множимо на 3, із одержаного добутку віднімаємо 3 і результат ділимо на 3 (чисельник завжди ділиться на 3, тому така дія можлива) і одержимо попереднє число за три операції.

Неважко одержати формулу оберненого переходу: якщо двічі помножити натуральне число на 3, із одержаного добутку 2*n*– 1 разів відняти 3 і результат розділити на 3, одержимо наступне натуральне число.

. (2.2)

Отже, наприклад, для переходу 81→82 використаємо одержану формулу (2.2), при цьому виконавши 164 операції (дві — множення на 3, сто шістдесят одну — віднімання 3 і одну – ділення на 3 ). Але можна навести багато інших прикладів даного переходу за іншими алгоритмами з меншою кількістю кроків. Наприклад, перехід за 24 кроки:

.

Тобто, обернений перехід за запропонованим алгоритмом не оптимальний.

Таким чином, було доведено, що перехід від числа до наступного чи до попереднього можливий. Але чи оптимальна кількість операцій указаних алгоритмів — це складне питання при такому підході в розв’язанні. Крім того, розв’язати задачу в загальному вигляді, для будь-яких чисел *n* і *m,* міркуючи аналогічно, досить важко. Питання існування оптимального алгоритму буде розглянуто у наступному розділі.

### Вибір комп’ютерних засобів для програмної реалізації

Бурхливий розвиток Інтернет-технологій та мережі Інтернет призвів до того, що інтернет-додатки стали більш актуальними, поширеними і все більш складними, граючи таким чином основну роль у більшості онлайн проектів. Зважаючи на переваги веб-орієнтованих додатків, саме такий формат було обрано для програмної реалізації розробленого алгоритму. Для цього було обрано такі технології, як HTML, CSS та JavaScript [7]. Розглянемо більш детально кожну з них та встановимо доцільність їх використання.

HTML — це стандартна мова розмітки веб-сторінок в Інтернеті. Більшість веб-сторінок створюються за допомогою мови HTML (або XHTML). Документ HTML оброблюється браузером та відтворюється на екрані у звичному для людини вигляді.

Використання HTML [8] забезпечує:

* створення структурованого документу шляхом позначення структурного складу тексту, такого як заголовки, абзаци, списки, таблиці, цитати та інше; отримання інформації із Всесвітньої мережі через гіперпосилання;
* створення інтерактивних форм;
* включення зображень, звуку, відео, та іншихоб'єктів до тексту.

Для того, щоб створити привабливий інтерфейс, можна використати каскадні таблиці стилів (англ. Cascading Style Sheets або скорочено CSS) — спеціальну мова, що використовується для відображення сторінок, написаних мовами розмітки даних. Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML для того, щоб визначити кольори, шрифти, верстку та інші аспекти вигляду сторінки. Одна з головних переваг — можливість розділити зміст сторінки (або контент, наповнення, зазвичай HTML, XML або подібна мова розмітки) від вигляду документу (що описується в CSS). Таке розділення може покращити сприйняття та доступність контенту, забезпечити більшу гнучкість та контроль за відображенням контенту в різних умовах, зробити контент більш структурованим та простим, прибрати повтори тощо. CSS має порівняно простий синтаксис і використовує небагато англійських слів для найменування різних складових стилю.

JavaScript — це динамічна, [об'єктно-орієнтована](http://uk.wikipedia.org/wiki/%D0%9E%D0%B1%27%D1%94%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D1%96%D1%94%D0%BD%D1%82%D0%BE%D0%B2%D0%B0%D0%BD%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)[мова програмування](http://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) [7]. Найчастіше використовується як частина [браузера](http://uk.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80), що надає можливість коду на стороні [клієнта](http://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D1%96%D1%94%D0%BD%D1%82-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80%D0%BD%D0%B0_%D0%B0%D1%80%D1%85%D1%96%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0) взаємодіяти з користувачем, керувати браузером, [асинхронно](http://uk.wikipedia.org/wiki/AJAX) обмінюватися даними з [сервером](http://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D1%96%D1%94%D0%BD%D1%82-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80%D0%BD%D0%B0_%D0%B0%D1%80%D1%85%D1%96%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0), змінювати [структуру](http://uk.wikipedia.org/wiki/%D0%9E%D0%B1%27%D1%94%D0%BA%D1%82%D0%BD%D0%B0_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B4%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0) та [зовнішній вигляд](http://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%B4%D0%B8%D0%B7%D0%B0%D0%B9%D0%BD)[веб-сторінки](http://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0). Мова JavaScript також використовується для [програмування](http://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) на стороні серверу (подібно до таких мов програмування, як [Java](http://uk.wikipedia.org/wiki/Java) і [C#](http://uk.wikipedia.org/wiki/C_Sharp)), розробки [ігор](http://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D1%96_%D1%96%D0%B3%D1%80%D0%B8), стаціонарних та мобільних додатків, сценаріїв в прикладному [ПЗ](http://uk.wikipedia.org/wiki/%D0%9F%D0%97), всередині [PDF](http://uk.wikipedia.org/wiki/PDF)-документів тощо. Отже, будемо використовувати JavaScript для написання скрипту, який буде виконувати розрахунок і виводити на екран результати. Тобто, саме скрипт буде містити логіку роботи веб-додатку. Вибір мови JavaScript для реалізації поставленої задачі зумовлений тим, що ця мова є досить простою та зрозумілою.

Використання вищезазначених технологій достатнє для реалізації поставлених задач. Крім цього, такий вибір забезпечує доступність додатку і зручність у застосуванні.

## РОЗДІЛ 3 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

### Побудова алгоритму розрахунків

Позначимо найменшу кількість операцій переходу від числа *m* до числа *n* як . Спочатку розглянемо кілька зауважень.

*Перше зауваження (відносно пріоритету дій).*

1. З’ясуємо, яку з операцій — множення чи віднімання першочергово «вигідніше» використовувати. Нехай для переходу від числа *m* ми спочатку помножили *y* разів на число 3, а потім відняли разів число 3, тобто:

. (3.1)

Тоді кількість операцій .

Аналогічно, якщо спочатку відняти разів число 3, а потім помножити *y* разів на число 3:

, (3.2)

то кількість операцій .

Із рівності правих частин слідує, що (так як). Отже, якщо є можливість зробити вибір між множенням на число 3 і відніманням числа 3, то перевагу треба надавати операції віднімання. У межах одного періоду треба спочатку відняти якомога більше 3, а вже потім множити на 3 скільки потрібно разів.

1. За допомогою аналогічних міркувань з’ясуємо «пріоритет» операцій ділення та віднімання:

, тоді .

, тоді .

, тоді .

Отже, якщо є можливість вибирати між діленням на три і відніманням трійки, то потрібно вибирати ділення.

**Висновок:** пріоритет операцій має бути такий — “ділення” ,”віднімання”, ”множення”.

*Друге зауваження (відносно однозначності).*

Перехід від *m* до *n* може бути неоднозначним при одній і тій же кількості операцій. Розглянемо приклади.

Приклад 1:

**27 →12**

Перехід:

а) 27 – 3 – 3 – 3 – 3 – 3 =12, ; б) – 3 – 3 = 12, .

Приклад 2:

**11 →28**

Перехід:

а)  = 28, ; б) , .

Проаналізуємо останній приклад. В обох варіантах дії виконуються в різній послідовності. У результаті:

a) перехід до *n* на останньому кроці 3*n*→*n*;

б) перехід до *n* на останньому кроці *n+*3→*n*, але перед цим було 3(*n*+ 3)→*n*+ 3.

Отже, якщо *m*< *n* і *n* не ділиться на 3, то обов’язково на якомусь кроці було множення на 3.

*Третє зауваження (виключення).*

Розглянемо деякі особливі випадки:

**1→2**; перехід: .

**1→3**; перехід: .

**2→3**; перехід: .

**2→1**; перехід: .

**3→1**; перехід: 

**3→2**; перехід: .

Далі будемо вважати, що *m* > 3. Розв’яжемо задачу у загальному вигляді.

Розглянемо можливі випадки.

**Нехай *m*< *n*.** Згідно другого зауваження: якщо *m* < *n* і *n* не ділиться на число 3, то обов’язково на якомусь кроці було множення на число 3. Якщо *n* не ділиться на 3, то можливо лише 3*n*→*n* (3*n*:3 = *n*).

Тому позначимо:

. (3.3)

Далі розглядатимемо перехід *m*→*n*0 .

Нехай , .

Введемо змінну *t*, .

Будемо множити *m* на степені трійки, щоб максимально наблизити до *n*0 (якщо при множенні попали в *n*0 — це ідеальний варіант). Продемонструємо ці дії на рисунку (рис.3.1). Наближення *m* до *n*0 має вигляд . Це число може «перескочити» за *n*0 ( > *n*0), а може наблизитися максимально зліва, тоді . Позначимо різницю (відстань) між правою точкою ( або ) та *n*0 як *p*.

Рис.3.1 — Визначення параметрів

Очевидно, що. При чому  тоді і тільки тоді, коли *n*0 = (*p* = 0).

Нехай:

. (3.4)

Тобто — відповідний степінь трійки. Тоді  *p* > 0, .

Задача зводиться до «подолання» цієї відстані за оптимальну кількість кроків.

У випадку подільності на 3, розглянемо можливі подання числа *n*0 і порівняємо значення відповідних їм.

1) *n*0 = , ; *n*0 = , ;

Тут  при . Тому вигіднішим є перший варіант.

1. При *c* > 1: , ; , ;

Очевидно, *c* > *a*, тоді *c*= *a* + *c*1 , *b*= *c*1∙*>c*1при.

Отже перед множенням на 3 вигідно відняти від *m* якомога більше трійок (в межах основного розряду: *m* – 3*c* > 3*α* ).

3) Якщо в результаті віднімання трійок здійснюється перехід до меншого розряду (*m* – 3*c* < 3*α*), значення  збільшується:

;

;

;



Тут, очевидно , тоді .

Розкладемо *р* по степеням трійки:

, , (3.5)



З іншого боку:

a) ;

б) .



Можливі ситуації:

1. **Перший випадок: .**

 (3.6)

. (3.7)

Тоді:

. (3.8)

Тут при **** рівність **** не можлива, оскільки тоді , але .

Приклад:

**8→33**

1. 33 ділиться на 3 без остачі.
2. .
3. 
4. .
5. .
6. *p* = 72 – 33 = 39.
7. 39 = 33 + 32 + 31 → *k* = 3
8. 
9. ((8 – 3∙1)∙3 – 3∙1)∙3 – 3∙1 = 33
10. **Другий випадок: .**

. (3.9)

. (3.10)

У конкретних ситуаціях можливе зменшення кількості операцій у залежності від подільності *m* на певний степінь трійки:

а) 

У такому випадку (3.10) можна подати у вигляді:

.

 (3.11)

Очевидно, що:

 (3.7) =  (3.8) + 2. (3.12)

Рівність можлива лише при , .

Приклади:

**30→273**

1. 273 ділиться на 3 без остачі.
2. .
3. .
4. .
5. .
6. *p* = 810 – 273 = 537.
7. 537 = 2∙35 + 1∙33 + 2∙32 + 2∙31 → *k* = 5.
8. .
9. .

б) 

Тут (3.11) можна подати у вигляді:

.

**.** (3.13)

Очевидно, що:

 (3.11) ≥  (3.13). Рівність лише при , 

Приклад:

**90→555**

1. 555 ділиться на 3 без остачі.
2. .
3. .
4. .
5. .
6. *p* = 810 – 555 = 255.
7. 255 = 35 + 32 + 31 → *k* = 5.
8. .
9. .

в) Розглядаючи інші випадки помічаємо, що при  маємо наступні формули:

; (3.14)

.

Зауважимо, що у випадку, коли *n* не ділиться націло на число 3, спочатку робимо перехід у 3*n*, а потім, діленням на число 3, у *n*.

Приклад:

**17→35**

Спочатку 17→105.

1. 105 ділиться на 3 без остачі.
2. .
3. .
4. .
5. .
6. *p* = 153 – 105 = 48.
7. 48 = 35 +2∙32 + 31 → *k* = 5.
8. . 
9. 105 = ((17 – 3)∙3 – 32)∙3 – 3∙1.

Таким чином, вирази (3.1)-(3.14) становлять шуканий алгоритм. Вони будуть покладені в основу програмної реалізації.

### Схема роботи та структура додатку

Розроблений додаток складається з трьох компонент: веб-сторінки index.html (див. Додаток А), файлу зі стилями main.css (Додаток Б) та файлу зі скриптом calculate.js (Додаток В), який виконує розрахунок. Сторінка (див. рис.3.2) містить інформацію про додаток, поля для введення параметрів *m* та *n*, чекбокс «Відображати формули», кнопку «Виконати розрахунки».

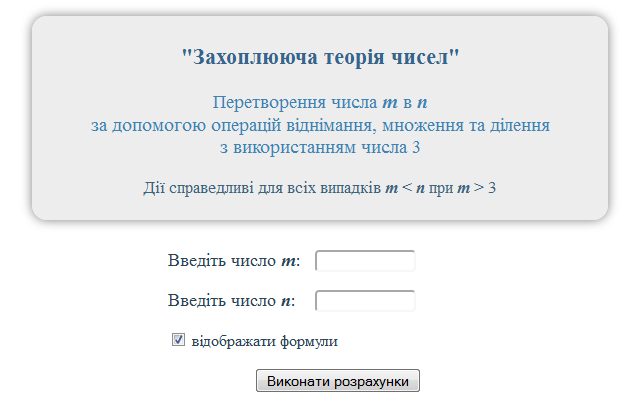


Рис. 3.2 — Інтерфейс сторінки index.html

Після натискання на кнопку «Виконати розрахунки» запускається скрипт, відбувається розрахунок та виведення результату на екран. По своїй структурі веб-сторінка представляє собою набір блоків (елементів div), які позиціоновані за допомогою каскадних таблиць стилів (CSS). Також за допомогою таблиць стилів визначений розмір блоків, формат тексту та інші візуальні ефекти.

З точки зору HTML сторінка представляє собою кілька блоків (елементів div), які зорієнтовані за допомогою відповідних стилів. Також визначений стиль для тексту кожного блоку, а також для окремих елементів, наприклад, для кнопки «Виконати розрахунки».

Основною частиною додатку є скрипт, адже він реалізує всю логіку роботи додатку і час алгоритм розрахунку. Він складається з декількох функцій, а саме:

* avtomat(): основна функція, яка реалізує розроблений алгоритм.
* alfabeta(aaa): допоміжна функція, за допомогою якої розраховується степінь, в який треба піднести трійку, щоб отримати число, що покриває задане число aaa. Використовувалась для розрахунку параметрів *α* та *β.*
* div(val,by): допоміжна функція для визначення цілої частини від ділення числа val на by. Зазвичай, в інших мовах ця функція стандартна, а в JavaScript вона відсутня, тому реалізована самостійно.
* delenie(element): допоміжна функція для визначення степеня, при якому ділення заданого числа element на трійку в цьому степені буде без залишку. Використовувалась для розрахунку параметру *w*.

Таким чином, основною функцією є avtomat(). Саме в ній реалізовано не лише розроблений алгоритм, а й зчитування введених користувачем вхідних параметрів, валідація цих параметрів, виведення результату на екран. Для доступу до html-елементів за допомогою JavaScript застосовувались такі методи, як getElementById, getElementsByTagName.

Принцип роботи скрипту представлений на схемі на рис. 3.3:

Зчитування параметрів *m* та *n*

Валідація введених значень *m* та *n*:

* *m* та *n* числа;
* *m* та *n* більше нуля;
* *m >* 3та *n >* 4;
* *m < n.*

Перевірка особливого випадку

*m =*3*xn,*

де *x* — будь-який степінь.

Перевірка та визначення *n*0 (3.3)

Розрахунок параметрів *α,* *β, t.*

Розрахунок параметрів *α,* *β, t*(рис.3.1)*, ν*(3.4)*, p* (3.5)*, k* (3.5), *w*.

Розклад числа *p*.

Перевірка умови *k* ≤ *ν* + 1

Перевірка умови *k* ≥ *ν* + 2

Розрахунок    
за формулою (3.8)

Розрахунок    
за формулою (3.10)

Перехід від *m* до *n*   
за формулою (3.6)

Перехід від *m* до *n*   
за формулою (3.9)

Виведення результату

Рис.3.3 — Схема роботи скрипту

### Тестування роботи додатку та аналіз результатів

Перш за все, слід перевірити роботу додатку при введенні невалідних даних, тобто в даному випадку, це текст, від’ємні значення, пусті поля та нульові значення. Якщо хоча б одна з цих умов хоча б для одного з полів виконується, то буде виведено вікно з повідомленням про необхідність введення валідних значень (див. рис. 3.4).

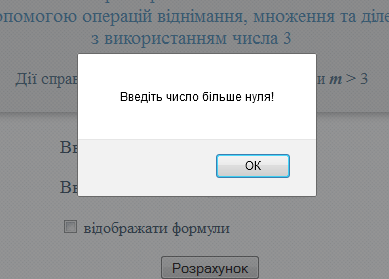


Рис.3.4 — Діалогове вікно з повідомленням про невірно введені дані

При введенні даних, що не відповідають умовам *n* > *m*, *n* ≤ 4, *m* ≤ 3, буде виведено вікно з відповідним повідомленням (див. рис.3.5).

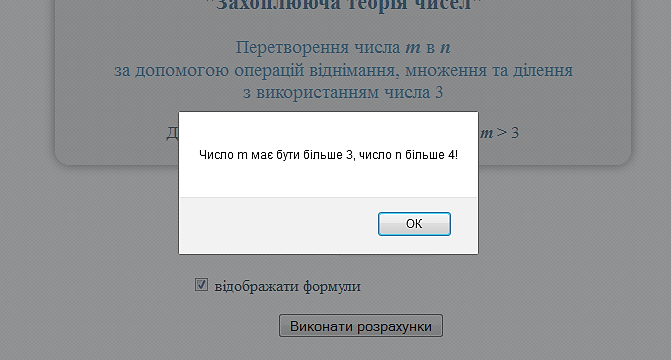


Рис.3.5 — Діалогове вікно з повідомленням про невідповідність даних початковим умовам

Тобто, така валідація даних забезпечує захист від невірного введення даних користувачем і отримання неправдивих результатів.

Крім цього, передбачена перевірка простих випадків, наприклад, якщо *n* ділиться на *m* у будь-якому степені, крім нульового. Якщо дана умова виконується, то алгоритм не запускається, а скриптом розраховується лише степінь трійки. Такий випадок представлений на рис. 3.6.

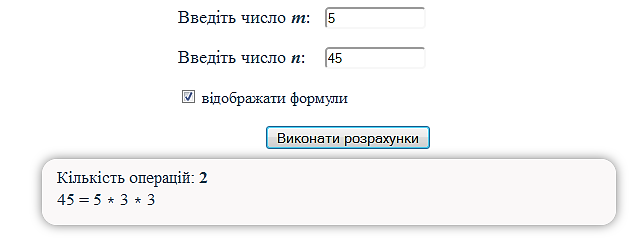


Рис. 3.6 — Результат роботи для випадку *m*= 5, *n*= 45

Протестуємо додаток для деяких прикладів, що були описані в попередніх розділах.

1. Перехід 8→33 (рис. 3.6).

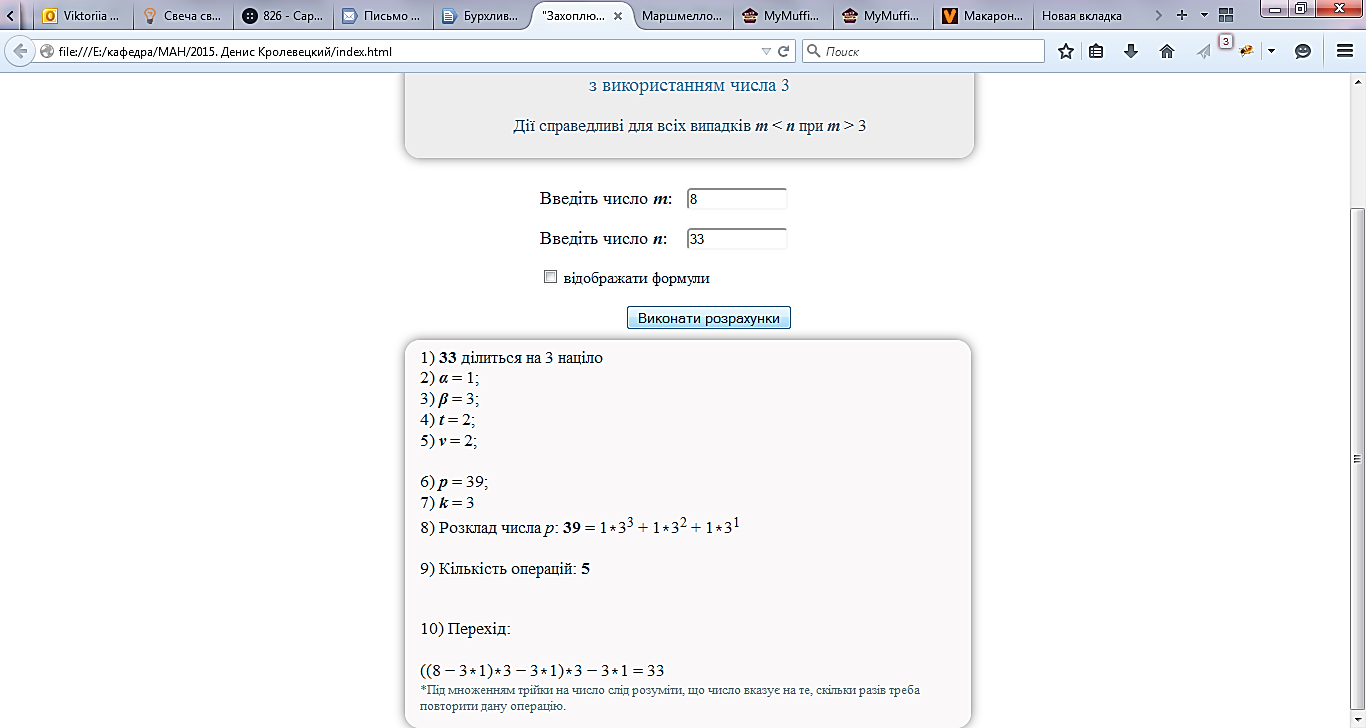


Рис. 3.6 — Результат роботи для випадку *m*= 8, *n*= 33

1. Перехід 30→273 (рис. 3.7).

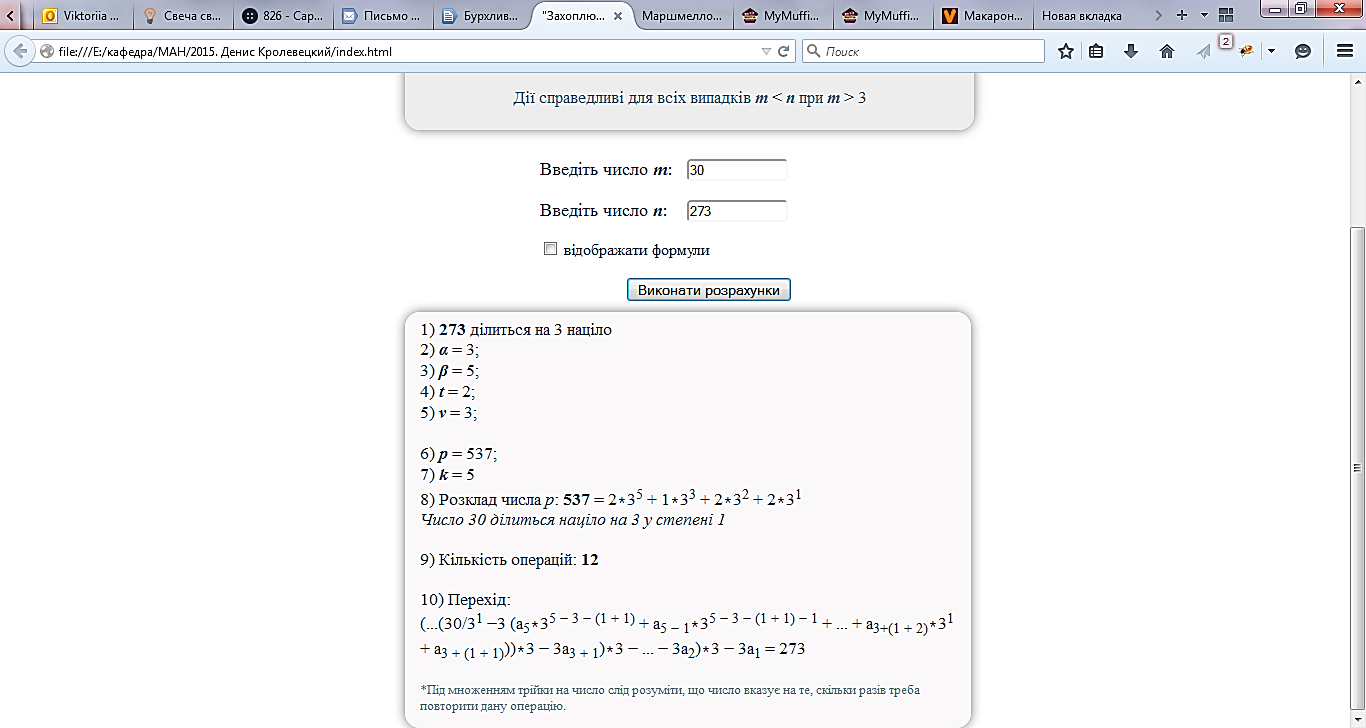


Рис. 3.7 — Результат роботи для випадку *m*= 30, *n*= 273

Отже, результати роботи додатку (рис.3.6-3.7) співпадають з результатами, отриманими в пункті 3.1. Це свідчать про те, що додаток працює правильно.

Крім виведення розрахунків, передбачена ще одна опція — виведення на екран формул, за якими проводились розрахунки. Щоб активувати цю опцію, треба відмітити чекбокс «Відображати формули» (дис. рис. 3.9).

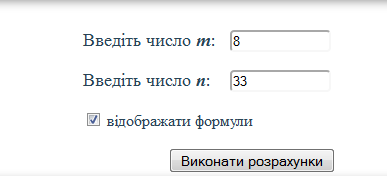


Рис. 3.8 — Активація опція відображення формул

Якщо опція відображення формул активна, тоді окрім проміжних результатів розрахунків відображаються формули. Такий результат показано на рис. 3.9.

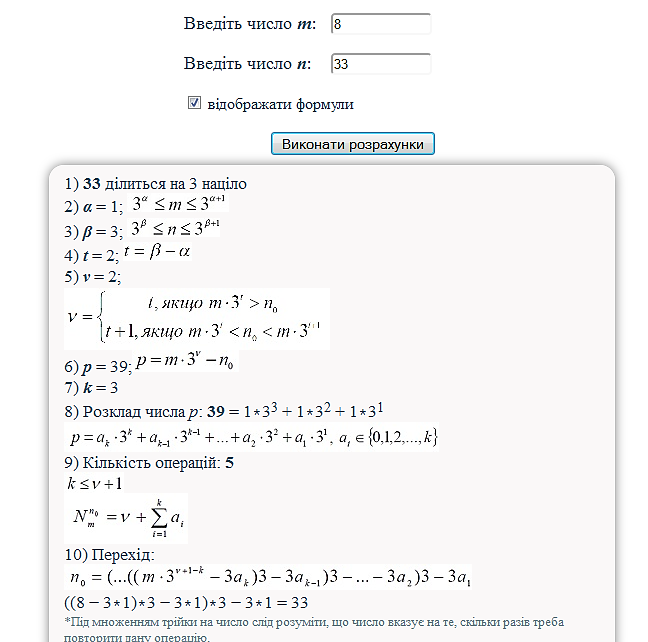


Рис. 3.9 — Виведення результату з формулами

Відображення формул дозволяє не просто отримати схему переходу в заданого числа *m* в число *n* оптимальним шляхом, а й зрозуміти принцип роботи всього алгоритму. Таким чином, цей додаток можна використовувати для навчання.

### ВИСНОВКИ

В результаті роботи було розроблено алгоритм оптимального переходу з заданого числа *m* в *n* за допомогою операцій віднімання, множення та ділення з використанням числа 3. Цей алгоритм був програмно реалізований у вигляді веб-орієнтованого додатку, який дозволяє не лише автоматизувати обчислення, а й виводити проміжні результати, формули, що дає змогу проводити аналіз та вивчати роботу алгоритму.

Для програмної реалізації були використані клієнтські веб-технології, такі як HTML, CSS та JavaScript. Це дало змогу реалізувати необхідний функціонал і зробити додаток незалежним від платформ та операційних систем.

Розроблений додаток був протестований шляхом порівняння результатів з раніше отриманими теоретичними результатами. Так як вони співпали, можна вважати,що додаток працює правильно.

Результати роботи можуть бути покладені в основу фундаментальних та прикладних досліджень в галузі математики, криптографії чи знайдуть застосування у тріскових комп’ютерах. Також розроблений додаток може бути використаний в навчальному процесі для залучення учнів до вивчення тем, що не входять в навчальну програм та для привернення їх уваги до математики.

### СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Алексеев А. Н. Проектирование для worldwideweb: учеб.пос. — Сумы : СумГУ, 2003. — 100 с.
2. Анісімов А. В. Алгоритмічна теорія великих чисел: Модулярна арифметика великих чисел. — К. : Академперіодика, 2001. — 153 с.
3. Назаренко А. М., Назаренко Л. Д. Тысяча и один пример. Равенства и неравенства: пособие для абитуриентов. — Слобожанщина, 1994. — 272 с.
4. Назаренко, А. М., Панченко Т. И. Математические кружки. Теория чисел. — Сумы : СумГУ, 2000. — 203 с.
5. Пасічник О. Г., Пасічник О. В., Стеценко І. В. Основи веб-дизайну: навч. посіб. — К. : Вид. група BHV, 2009. — 336 с.
6. Проценко О. Б. Web-програмування та web-дизайн. Технологія XML: навч. посіб. — Сумы : СумГУ, 2009. — 127 с.
7. Ташков П. А. Веб-мастеринг на 100%: HTML, CSS, JavaScript, PHP, CMS, AJAX, раскрутка — 2-е изд., испр. — СПб. : Питер, 2010. — 512 с.
8. Хейз Д. Освой самостоятельно HTML и XHTML: 10 минут на урок — 3-е изд.,. — М. : ИД "Вильямс", 2002. — 224 с.

ІНТЕРНЕТ-РЕСУРСИ

<http://uk.wikipedia.org/>

<http://htmlbook.ru/>

<http://www.w3schools.com/>

<http://shodennik.ua/>

<http://www.wolframalpha.com>

<http://osvita.ua/school/lessons_summary/proftech/34663/>

<http://teacher.at.ua/publ/innovacijni_tekhnologiji_navchannja/multimedijni_sistemi_navchannja_jak_novij_metodologichnij_zasib_interaktivnogo_navchannja_na_urokakh/63-1-0-15092>

### ДОДАТОК А

<html>

<head>

<meta charset="utf-8">

<title>"Захоплююча теорія чисел". Автор: Кролевецький Денис</title>

<link rel="stylesheet" href="main.css">

<script src="calculate.js">

</script>

</head>

<body>

<div id="header">

<p id="hed0">"Захоплююча теорія чисел"</p>

<p id="hed1">Перетворення числа <b><i>m</i></b> в <b><i>n</i></b> <br>за допомогою операцій віднімання, множення та ділення <br>з використанням числа 3</p>

<p id="hed2">Дії справедливі для всіх випадків <b><i>m</i></b> &lt;<b><i> n</i></b> при <b><i>m </i></b>> 3</p>

</div>

<br>

<div class="maindiv">

<p>Введіть число <b><i>m</i></b>: &nbsp; <input class="in" id="first" type="text" name="first" size="12" maxlength="7"></p>

<p>Введіть число <b><i>n</i></b>: &nbsp; &nbsp;<input class="in" id="last" type="text" name="last" size="12" maxlength="7"></p>

<p style="font-size:90%"><input type="checkbox" checked name="picture" id="picture"> відображати формули</p>

<p><input id="calculate" type="button" name="but1" onclick="avtomat()" value="Виконати розрахунки"></p>

</div>

<div id="rez" style="display:none;"></div>

</body>

</html>

### ДОДАТОК Б

#hed0 {

color: #36648B;

font-size:140%;

font-weight: bold;

text-align: center;

}

#hed1 {

color: #3F83B2;

font-size:120%;

text-align: center;

}

#hed2 {

color: #3E637C;

font-size:105%;

text-align: center;

}

#header {

border: solid 0px #817D7D;

background: #EEEDED;

position: absolute;

top: 3%;

left: 30%;

padding: 5px;

padding-left: 15px;

padding-right: 15px;

border-radius: 15px 15px 15px 15px;

box-shadow: 0 0 10px rgba(0,0,0,0.5);

width: 40%;

}

.maindiv {

position: absolute;

top: 36%;

left: 40%;

font-size: 110%;

color:#294457;

}

#calculate {

font-size: 80%;

position: relative;

left: 35%;

}

#calculate:hover{

color: #1B374B;

}

.in{

border-radius: 4px 4px 4px 4px;

}

#rez {

border: solid 0px #817D7D;

background: #FAF8F8;

position: absolute;

top: 62%;

left: 30%;

width: 42%;

-moz-box-sizing: border-box; /\* Для Firefox \*/

-webkit-box-sizing: border-box; /\* Для Safari и Chrome \*/

box-sizing: border-box; /\* Для IE и Opera \*/

padding: 8px;

padding-left: 15px;

padding-right: 15px;

padding-bottom: 15px;

border-radius: 15px 15px 15px 15px;

box-shadow: 0 0 10px rgba(0,0,0,0.5);

color:#294457;

font-size: 105%;

}

.vis{

display: none;

}

### ДОДАТОК В

function alfabeta(aaa){

var k=0;

do

{

r=aaa/3;

aaa=r;

k=k+1;

}

while (r>=1)

return k;

}

function div(val,by){

return (val-(val%by))/by;

}

function delenie(element){

if (element%3==0) {

w=1

u=0;

while (element % Math.pow(3,w)==0)

{

element=element/3;

u=u+1;

}

}

else u=0;

return u;

}

function avtomat() {

mtemp=document.getElementById("first");

ntemp=document.getElementById("last");

m=Number(document.getElementById("first").value);

n=Number(document.getElementById("last").value);

rezult=document.getElementById("rez");

if (isNaN(m)|isNaN(n)|n<=0|m<=0)

{ alert ("Введіть число більше нуля!")}

else if (mtemp.length>8 | ntemp.length>8)

{alert ("Введіть числа меньше 8 знаків")}

else if (m<=3 | n<=4)

{alert ("Число m має бути більше 3, число n більше 4!")}

else if (m>=n)

alert("Введіть правильне значення m та n!")

else if (n/m==3) {

document.getElementById('rez').style.display ="";

rezult.innerHTML="Помножте <b>"+m+"</b> на <b>3</b> для того, щоб отримати число <b>"+n+"</b>"

}

else if (delenie(n/m)>1){

koloper=delenie(n/m);

rezult.innerHTML="Кількість операцій: <b>"+koloper+"</b><br>"+n+" = "+m

for (o=1; o<koloper+1;o++){

rezult.innerHTML=rezult.innerHTML+" &lowast; 3"

}

}

else {

document.getElementById('rez').style.display ="";

if ((n % 3)==0) {

n0=n;

rezult.innerHTML="1) <b>"+n+"</b> ділиться на 3 націло<br>"

}

else {

n0=3\*n;

rezult.innerHTML="1) <b>"+n+"</b> ділиться на 3 з залишком<br>"

}

alfa=alfabeta(m)-1;

rezult.innerHTML=rezult.innerHTML+"2) <b><i>&alpha;</i></b> = "+alfa+"; <img src='img/alpha.jpg'><br>";

beta=alfabeta(n0)-1;

rezult.innerHTML=rezult.innerHTML+"3) <b><i>&beta;</i></b> = "+beta+"; <img src='img/beta.jpg'><br>";

t=beta-alfa;

rezult.innerHTML=rezult.innerHTML+"4) <b><i>t</i></b> = "+t+"; <img src='img/t.jpg'><br>";

tempval=m\*Math.pow(3, t);

nu=0;

if (tempval>=n0)

nu=t;

else

nu=t+1;

rezult.innerHTML=rezult.innerHTML+"5) <b><i>&nu;</i></b> = "+nu+";<br><img src='img/nju.jpg'><br>";

p=(m\*Math.pow(3, nu))-n0;

rezult.innerHTML=rezult.innerHTML+"6) <b><i>p</i></b> = "+p+";<img src='img/p.jpg'><br>";

k=alfabeta(p)-1;

rezult.innerHTML=rezult.innerHTML+"7) <b><i>k</i></b> = "+k+"<br>";

rezult.innerHTML=rezult.innerHTML+"8) Розклад числа <i>p</i>: <b>"+p+"</b> = ";

tempk=0;

sum=0;

i=0;

maskoef=[];

for (j=0; j<alfabeta(p); j++)

{

maskoef[j]=0;

}

do {

tempk=alfabeta(p)-1;

koef=div(p, Math.pow(3,tempk));

maskoef[tempk]=koef;

sum=sum+koef;

rezult.innerHTML=rezult.innerHTML+koef+"&lowast;3"+"<sup>"+tempk+"</sup>";

p=p-koef\*Math.pow(3,tempk);

if (p>0) rezult.innerHTML=rezult.innerHTML+" + "

}

while (p>0)

kol=0;

if (k<=nu+1) {

kol=nu+sum;

if ((n % 3)!=0)

{kol=kol+1}

else kol=kol;

rezult.innerHTML=rezult.innerHTML+"<br><img src='img/rozklad.jpg'><br>9) Кількість операцій: <b>"+kol+"</b>";

}

else {

w=delenie(m);

rezult.innerHTML=rezult.innerHTML+"<br><i>Число "+m+" ділиться націло на 3 у степені "+w+"</i>";

s1=0;

s2=0;

for (i=1; i<nu+2; i++)

{

s1=s1+maskoef[i];

}

j=k

while (j>=nu-w && j-nu-(w+1)>=0)

{

s2=s2+maskoef[j]\*Math.pow(3,j-nu-(w+1));

j=j-1;

}

kol=nu+2\*w+s1+s2;

rezult.innerHTML=rezult.innerHTML+"<br><img src='img/rozklad.jpg'><br>9) Кількість операцій: <b>"+kol+"</b>";

}

if (k<=nu+1){//случай k<=nyi+1

rezult.innerHTML=rezult.innerHTML+"<br><img src='img/k\_menshe.jpg'><br><img src='img/kol\_k\_menshe.jpg'><br>10) Перехід: <br><img src='img/rozklad\_k\_menshe.jpg'><br>";

kolskob=k-1;

if ((n % 3)!=0) {kolskob=kolskob+1}

for (i=1;i<kolskob+1;i++){

rezult.innerHTML=rezult.innerHTML+"("

}

power=nu+1-k;

if (power>0)

{rezult.innerHTML=rezult.innerHTML+m+"\*3<sup>"+power+"</sup>"}

else {rezult.innerHTML=rezult.innerHTML+m}

i=k;

while (i>1){

rezult.innerHTML=rezult.innerHTML+" &minus; "+"3&lowast;"+maskoef[i]+")&lowast;3"

i=i-1;

}

if (maskoef[1]>0)

{rezult.innerHTML=rezult.innerHTML+" &minus; 3&lowast;"+maskoef[1]}

if ((n % 3)!=0){

rezult.innerHTML=rezult.innerHTML+")/3";

}

rezult.innerHTML=rezult.innerHTML+" = "+n;

}

else {//k>=nju+2

rezult.innerHTML=rezult.innerHTML+"<br><img src='img/k\_bolshe.jpg'><br>10) Перехід: <br>";

rezult.innerHTML=rezult.innerHTML+"(...("+m+"&#47;3<sup>"+w+"</sup>&nbsp;&minus;3&nbsp;(a<sub>"+k+"</sub>&lowast;3<sup>"+k+" &minus; "+nu+" &minus; ("+w+" + 1)</sup> + a<sub>"+k+" &minus; 1</sub>&lowast;3<sup>"+k+" &minus; "+nu+" &minus; ("+w+" + 1) &minus; 1</sup> + ... + a<sub>"+nu+"+("+w+" + 2)</sub>&lowast;3<sup>1</sup> + a<sub>"+nu+" + ("+w+" + 1)</sub>))&lowast;3 &minus; 3a<sub>"+nu+" + " +w+"</sub>)&lowast;3 &minus; ... &minus; 3a<sub>2</sub>)&lowast;3 &minus; 3a<sub>1</sub>";

rezult.innerHTML=rezult.innerHTML+" = "+n+"<br><img src='img/rozklad\_k\_bolshe.jpg'>";

}

rezult.innerHTML=rezult.innerHTML+"<br><font style='font-size:80%;' color='#68838B'>\*Під множенням трійки на число слід розуміти, що число вказує на те, скільки разів треба повторити дану операцію.</font>"

}

if (!picture.checked) {

for (i=0; i<30; i++){

document.getElementsByTagName('img')[i].classList.add('vis');

}

}

}