## Word

Потрібно порахувати кількості входжень символів у текст-зразок і зашифрований текст, і упорядкувати символи обох текстів за цими кількостями. Літери, що розташовані в цих упорядкованих списках на однакових місцях, ПРИБЛИЗНО складатимуть ключ. Щоб віднайти ключ точно, треба подивитися, які слова розшифровано з помилками й внести відповідні зміни.

## Excel

Вартість мінімальних шляхів від лівої верхньої клітинки до клітинок верхнього рядка й лівого стовпця дорівнює сумам чисел від лівої верхньої клітинки до поточної. Для всіх інших клітинок керуємося таким принципом: вартість мінімального шляху до заданої клітинки дорівнює мінімуму від вартості оптимальних шляхів до клітинок зліва та зверху + значення в самій клітинці.

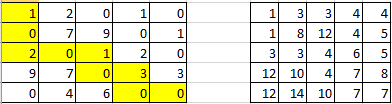
Щоб зафарбувати оптимальний шлях, можна застосувати умовне форматування, перед цим заповнивши в окремому діапазоні всі клітинки, що підлягають зафарбуванню, одним символом, наприклад ’1’, а всі інші клітинки — іншим, наприклад ‘0’. Внутрішня клітинка Х підлягає зафарбуванню у двох випадках:

* якщо клітинку справа вже заповнено символом ‘1’ і значення у клітинці, що відповідає Х у таблиці вартостей шляхів, менше або дорівнює значенню у клітинці справа згори (тоді з Х ми могли прийти в клітинку справа);

або

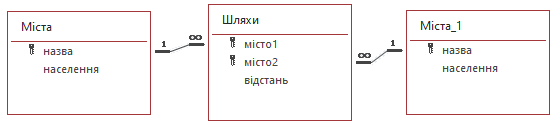
* якщо клітинку знизу вже заповнено символом ‘1’ і значення у клітинці, що відповідає Х у таблиці вартостей шляхів, менше або дорівнює значенню у клітинці зліва знизу (тоді з Х ми могли прийти в клітинку знизу).

Для клітинок на межі таблиці правила зафарбування будуть інші й простіші.



## Access

Зображена нижче схема бази даних (Міста та Міста\_1 — два подання тієї самої таблиці) гарантує, що:



1. Кількість населення для кожного міста можна задати тільки один раз.
2. Між містами А і В може існувати не більше 2 шляхів: від А до В та від В до А.

Завдяки особливостям відбору даних для списків форми користувач не зможе ввести два шляхи між тими самими містами.

Наприклад, вміст списку 1 у формі може формуватися на основі такого запиту:

SELECT назва

FROM Міста

WHERE isnull ([Forms]![Шляхи].[м2]) or назва not in (select місто2 from Шляхи where місто1=[Forms]![Шляхи].[м2]) and Міста.назва<>[Forms]![Шляхи].[м2] and назва not in (select місто1 from Шляхи where місто2=[Forms]![Шляхи].[м2]);

Тут [Forms]![Шляхи].[м2] — значення, що вибране в даний момент в іншому списку форми. А загалом запит читається так:

Вибрати назви таких міст, що:

в іншому списку не вибрано жодного значення (тоді в цьому списку відображатимуться всі міста);

**або**

(

у це місто не існує шляхів місто1->місто2 від міста1, вибраного в іншому списку;

**і**

назву цього міста не вибрано в іншому списку;

**і**

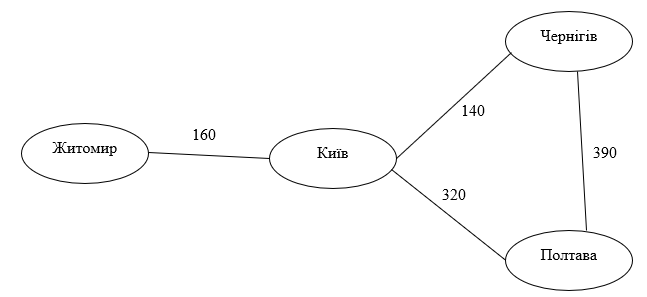
у це місто не існує шляхів місто2->місто1 від міста2, вибраного в іншому списку;

).

У цьому та в усіх інших запитах враховується, що якщо в таблиці шляхів є шлях A->B, то це означає, що є також шлях B->A тієї самої довжини, хоча в таблиці шляхів його не вказано.

Форма має оновлюватися за допомогою команди «Оновити» макросу, пов’язаного з подією «Поточний елемент».

Схема автошляхів, згідно з умовою задачі, є такою:



А результати запитів такими:

Запит а

|  |  |
| --- | --- |
| ввід користувача | результат |
| Житомир | Київ |
| Київ | Житомир, Чернігів, Полтава |
| Чернігів | Полтава, Київ |
| Полтава | Київ, Чернігів |
| Запит б |  |
| ввід користувача | результат |
| Житомир | 3000000 |
| Київ | 823000 |
| Чернігів | 3280000 |
| Полтава | 3283000 |
| Запит в |  |
| ввід користувача | результат |
| Житомир | Київ, 160 |
| Київ | Чернігів, 140 |
| Чернігів | Київ, 140 |
| Полтава | Київ, 340 |
| Запит г |  |
| ввід користувача | результат |
| Житомир | Житомир, Полтава, Чернігів |
| Київ | Київ, Полтава, Чернігів |
| Чернігів | Житомир, Київ, Чернігів, Полтава |
| Полтава | Житомир, Київ, Чернігів, Полтава |