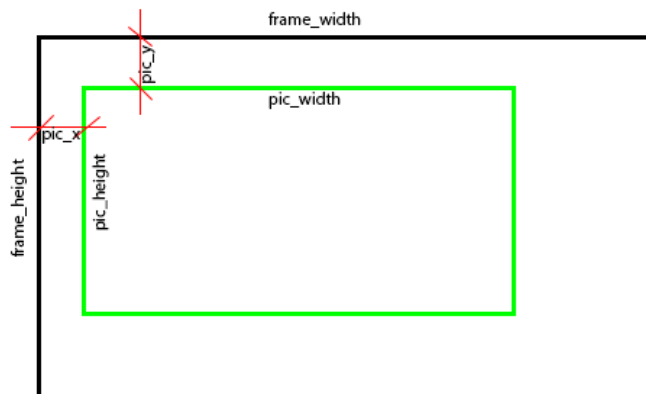


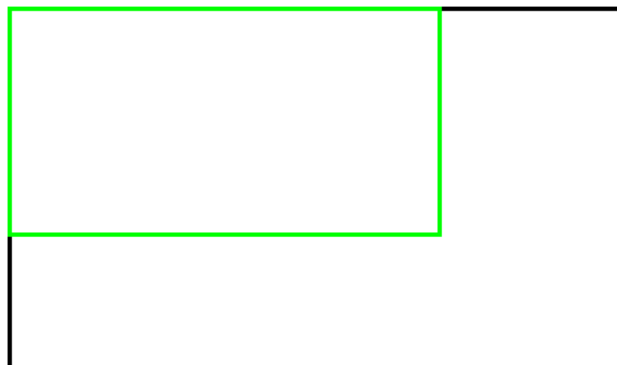
Завдання виконуйте будь-якою зручною вам мовою програмування (краще мовою з C/C++ подібним синтаксисом, але не обов'язково) без використання будь-яких сторонніх бібліотек або нестандартних розширень компілятора. Код має бути кросплатформним. Заборонено використовувати чужий код (друг, сусід, інтернет тощо). Усі завдання потрібно виконати самостійно. Оцінюватимуться вміння слідувати ТЗ, самостійно перевіряти результати своєї роботи, якість коду та використані підходи. Усього **3 завдання**. Третє завдання – опціональне і його можна не робити. Але якщо ви його зробите, це значно збільшить ваші шанси на успішне проходження тестового завдання:

1. Дано масив розміром `frame_width` x `frame_height`, де кожен елемент займає 4 байти (це просто масив з чисел, кожен з яких займає 4 байти, на практиці це може бути, наприклад, зображення). Масив заповнюється випадковими значеннями. Усередині цього масиву знаходяться "корисні дані" розміром `pic_width` x `pic_height`. Ці дані знаходяться на відстані `pic_x` від лівого краю масиву та `pic_y` від верхнього краю масиву (0-based, зображення Source). Все що знаходиться між чорним та зеленим прямокутниками – це просто "сміття" і нам не цікаве. Початок відліку – верхній лівий кут масиву. Необхідно написати функцію, яка скопіює "корисні дані" в лівий верхній кут цього ж масиву (на початок) (зображення Destination). Відобразіть на екрані вміст початкового та перетвореного масивів. Обов'язково використовуйте вказані на зображенні Source змінні для вирішення цього завдання та не використовуйте додаткові масиви. Значення `frame_width`, `frame_height`, `pic_width`, `pic_height`, `pic_x` та `pic_y` повинні вводитися з клавіатури.

Source



Destination



2. Написати програму, яка у циклі приймає число та виводить на екран строку прописом із сумою в доларах. Максимальне число – 1 мільярд.
Наприклад:
Вводимо — 1.00
Результат — "one dollar"
Вводимо — 123
Результат — "one hundred twenty-three dollars"
Вводимо — 1234567.89
Результат — "one million two hundred thirty-four thousand five hundred sixty-seven dollars eighty-nine cents".
Зверніть увагу на правильне використання символів пробіл " " і тире "-".
Приклад виводу для цілих чисел можна переглянути у пошуку Google "число=english", наприклад: 1234567=english.
Числа повинні вводитись з клавіатури.

3. Опціональне завдання. Як вхідні та вихідні джерела даних у цьому завданні повинні використовуватися файли. Приклади вхідних файлів лежать поруч із цим PDF.

Завдання.

Дотримуючись вказівок на старій карті, ви натрапили на секретну скарбницю Пірата! Скарбниця складається з **N** замкнених скринь, кожна з яких може бути відкрита тільки ключем певного типу. Крім того, якщо ключ використовується для відкриття скрині, його вже не можна використовувати знову. Всередині кожної скрині, крім безлічі скарбів, ви також можете знайти один або кілька ключів, які можна використовувати для відкриття інших скринь. Скриня може містити кілька ключів одного типу, і ви можете мати будь-яку кількість ключів.
Ви вже маєте принаймні один ключ, і на вашій карті видно, які інші ключі можна знайти в різних скринях. Маючи всю цю інформацію, чи можете ви з'ясувати, як відкрити всі скрині?

Наприклад, припустимо, що в скарбниці чотири скрині, як описано нижче, і ви почали з одного ключа типу 1:

Номер скрині	Тип ключа для відкриття скрині	Типи ключів усередині скрині
1	1	Нема
2	1	1, 3
3	2	Нема
4	3	2

У цьому прикладі ви можете відкрити всі скрині, якщо відкривати їх у наступному порядку: 2, 1, 4, 3. Якщо ви почнете з того, що спочатку відкриєте скриню № 1, то ви витратите свій єдиний ключ і застрягнете.

Вхідні дані.

У першому рядку вхідних даних вказано кількість тестових випадків **T**. Далі йдуть **T** тестових випадків. Кожен тестовий випадок починається з одного рядка, що містить два позитивні цілих числа **K** і **N**, що представляють кількість ключів, з якими ви починаєте, і кількість скринь, які вам потрібно відкрити.

Далі слідує рядок, що містить **K** цілих чисел, що представляють типи ключів, з якими ви починаєте.

Після цього буде **N** рядків, кожен з яких представляє одну скриню. Кожен рядок починається з цілих чисел **T_i** та **K_i**, що вказують тип ключа, необхідний для відкриття скрині, та кількість ключів усередині скрині. За цими двома числами будуть ще **K_i** цілих чисел, що вказують типи ключів, що містяться в скрині.

Результат.

Для кожного тестового випадку виведіть один рядок, що містить "Case #x: C₁ C₂ ... C_N", де x - номер випадку (починаючи з 1), а C_i - індекс (починаючи з 1) i-ї скрині, яку ви повинні відкрити.

Якщо є кілька способів відкрити всі скрині, оберіть «лексикографічно найменший» спосіб. Іншими словами, ви повинні зробити C₁ якомога меншим, а якщо є кілька способів зробити C₁ якомога меншим, виберіть той варіант, який робить C₂ якомога меншим, і так далі.

Якщо немає можливості відкрити всі скрині, ви повинні замість цього вивести один рядок, що містить "Case #x: IMPOSSIBLE".

Обмеження.

$$1 \leq T \leq 25.$$

$$1 \leq K.$$

Усі типи ключів будуть цілими числами від 1 до 200 включно.

Набір вхідних даних.

$$1 \leq N \leq 200.$$

У кожному тестовому випадку буде максимум 400 ключів.

Приклад.

Вхідні дані	Результат
3 1 4 1 1 0 1 2 1 3 2 0 3 1 2 3 3 1 1 1 1 0 1 0 1 0 1 1 2 1 1 1	Case #1: 2 1 4 3 Case #2: 1 2 3 Case #3: IMPOSSIBLE