

Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет інформатики та обчислювальної техніки

Кафедра автоматики та управління в технічних системах

Лабораторна робота №8

**Алгоритми та структури даних – 1. Основи алгоритмізації**

*Тема: «Машина Тюрінга»*

Команда №6

|  |  |  |
| --- | --- | --- |
| Виконали  студенти групи ІТ-03: |  | Перевірив: |
|  |  |  |
| Чабан А.Є. |  |  |
| Філянін Н.С. |  |  |
| Хамад І.А. |  | ас. Цимбал С. І. |

Київ 2020

Мета роботи: ознайомитися з симулятором машини Тюрінга та навчитись

використовувати його для розв’язування простих задач.

Хід роботи:

Необхідно запрограмувати машину Тюрінга враховуючи умови:

˗ машина починає роботу з положення на першому символі першого слова на

стрічці;

˗ після виконання роботи вхідні слова залишаються на стрічці, а відповідь

дописується за останнім вхідним словом через порожню клітинку;

˗ вхідні параметри записуються в бінарному поданні і розділяються

порожньою клітинкою на стрічці.

В таблиці 8.1 наведено варіанти завдань для виконання на симуляторі машини

Тюрінга, формула обрахунку – No варіанту = No команди % 7 + 1.

В симуляторі завантажено простий приклад з поясненнями для розуміння

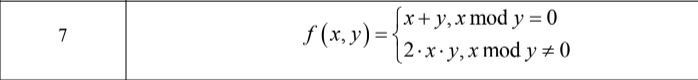
роботи машини Тюрінга загалом та синтаксису заданої машини.

Варто зазначити, що для всіх варіантів вхідні числа x, y є невід’ємними цілими

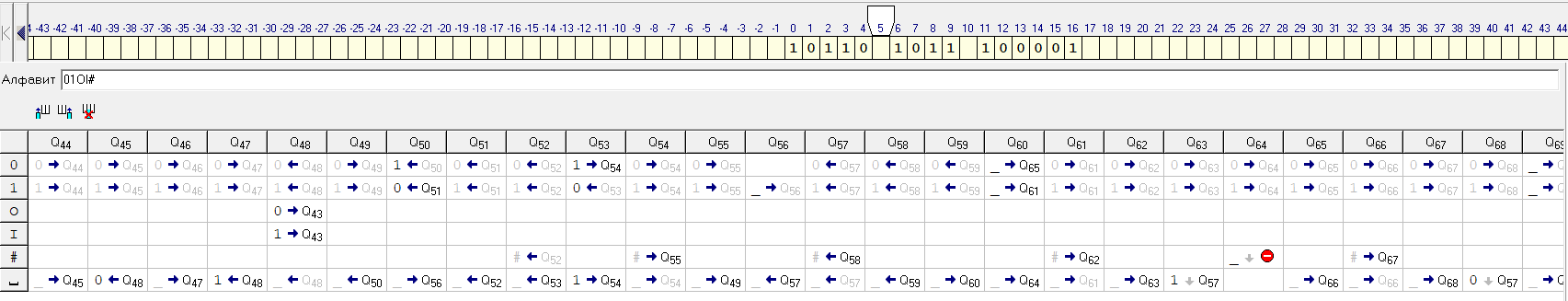
числами (x > 0, y > 0).

[\* Завантажити файл МТ можна на GitHub \*](https://github.com/it03team6/asdlabs/tree/master/asdlab8)

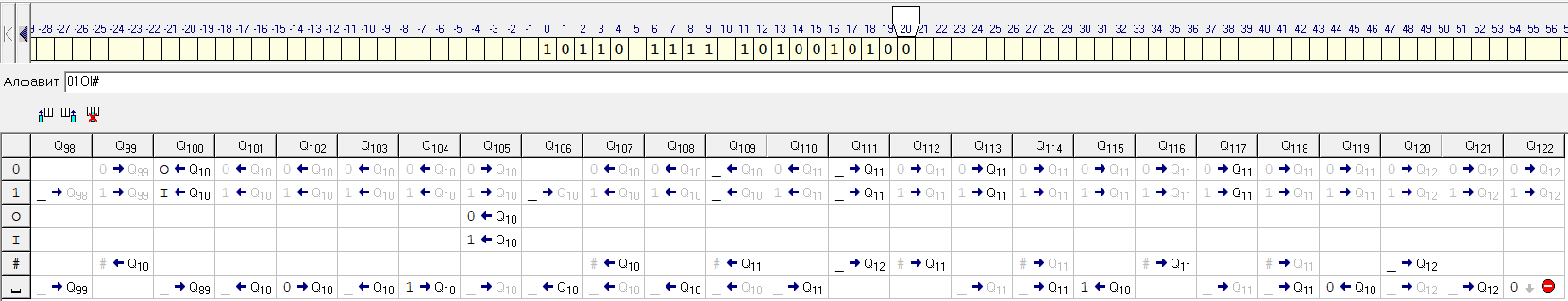
Завдання:



Результат виконання якщо x%y=0 (x=22,y=11):

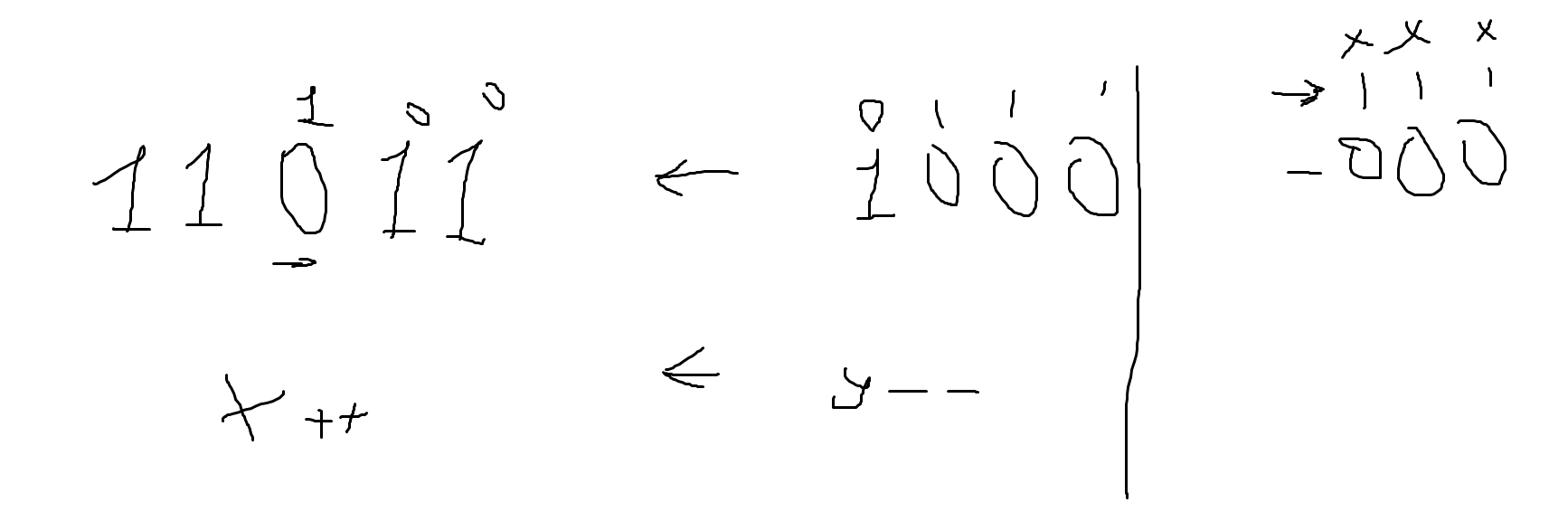
Як бачимо — результат вийшов 100001 (33). Що показує правильність роботи в данному випадку, теперь візьмемо випадок коли x%y!=0

Нехай х=22, у=15

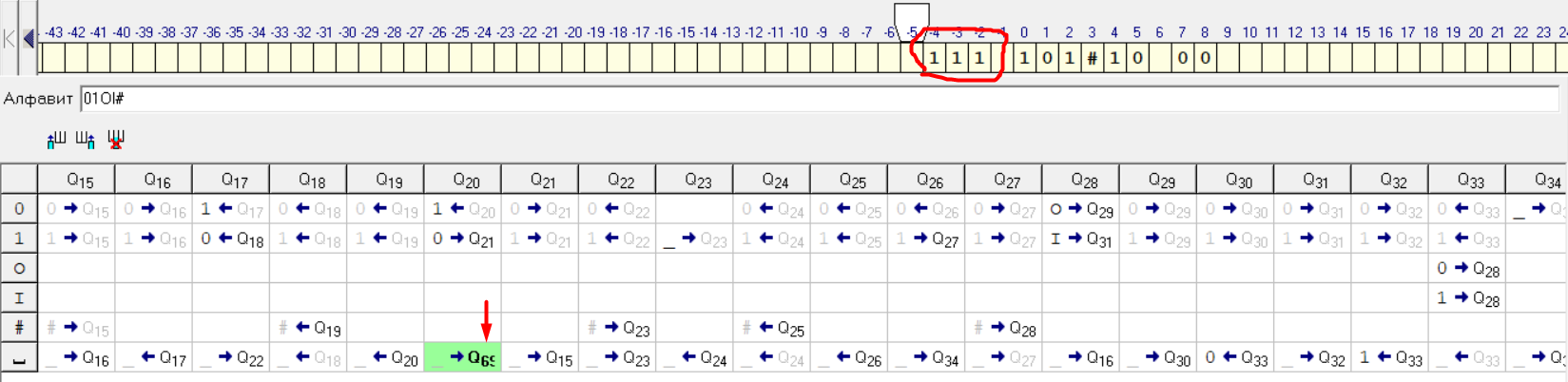
Як бачимо в результаті отримуємо 660. Як і має бути при виконанні формули 2\*x\*y.

Спершу перевіримо чи ділиться x на у, і для цього створимо їх копії і будемо працювати лише з копіями, бо при виконанні дій ми будемо їх змінювати, що призведе до втрати інформації. Для створення кордонів між х та у, використаємо додатковий символ «#».

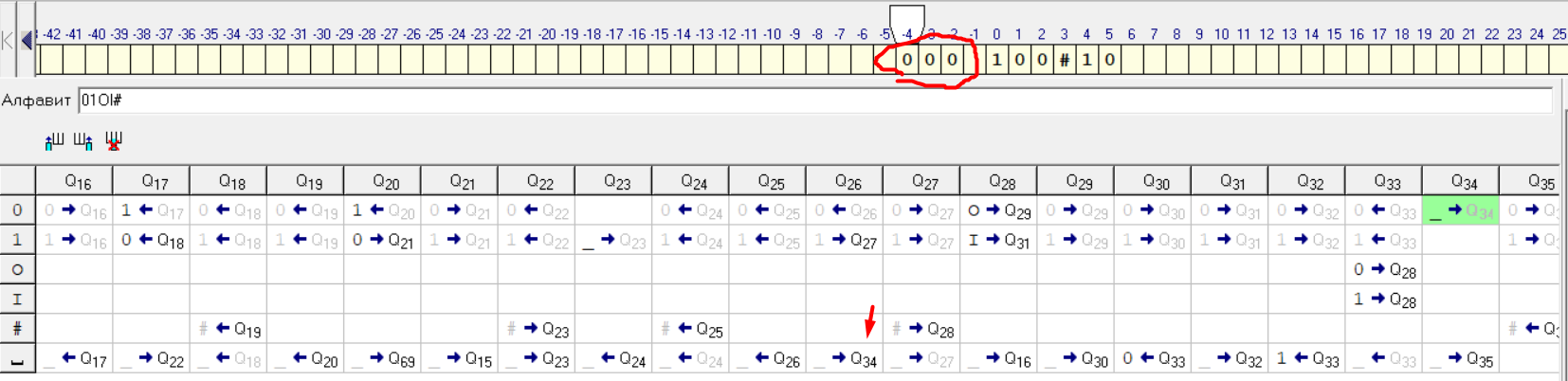
Всі функції реалізовані шляхом рекурсивного виклику їх самих та інкревентування (при додаванні та множенні) та декрементування (при відніманні та діленні).

.

(кривенька ілюстрація алгоритму). Тобто якщо якщо з першого разу число не віднялося (вся копія х не стала нулями) — ми рекурсивно викликаємо повторне віднімання. І якщо х таки стає нулями — це означає що число таки ділиться націло і далі ми виконує додавання (за аналогічним алгоритмом). Якщо ж ні, алгоритм замінить в числі всі 0 на одиниці, а дільник все ще не закінчиться, дойде то до пропуску і зрозуміє що число не ділиться націло і перейде до виконання функції множення.

.

А якщо отримуємо нулі — переходимо до додавання.

Висновок: під час виконання данної лабораторної роботи наша команда ознайомилася та навчилася використовувати машину Тьюринга та виконувати на ній прості алгоритми.