|  |  |  |  |
| --- | --- | --- | --- |
| **МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  **КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені Тараса Шевченка**  **ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**  **Кафедра програмних систем і технологій**  **Дисципліна**  **«Кросплатформне програмування»**  **Лабораторна робота 2** | | | |
| **Виконав:** | Чушенко Ярослав Володимирович | **Перевірив**: | Петрівський В.Я |
| Група | ІПЗ-41 | Дата перевірки |  |
| Форма навчання | денна | Оцінка |  |
| Спеціальність | 121 |
| 2023 | | | |

**Варіант – 54**Репозиторій на Github: [Посилання](https://github.com/PingvinAustr/CrossplatformKNU)  
 **Умова завдання:**Дано натуральне число N. Над ним можна провести наступний набір операцій:

* віднімати одиницю;
* ділити втричі, якщо число кратно трьом;
* ділити на два, якщо число парне.

Після виконання однієї з операцій до отриманого результату можна застосувати зазначені операції, і робиться це до тих пір, поки результат не виявиться рівним 1.

*Вхідні дані:*

Вхідний файл INPUT.TXT містить натуральне число N (N ≤ 106).

*Вихідні дані:*

У вихідний файл OUTPUT.TXT виведіть найменшу кількість операцій, в результаті яких буде отримана одиниця.

*Приклади:*

|  |  |  |
| --- | --- | --- |
| № | INPUT.TXT | OUTPUT.TXT |
| 1 | 5 | 3 |
| 2 | 1 | 0 |
| 3 | 10 | 3 |

**Виконання роботи:**

1. **Код:**
2. using System.Diagnostics;
3. class Program
4. {
5. static void Main()
6. {
7. int n = int.Parse(File.ReadAllText("..//..//..//INPUT.TXT").Trim());
8. Console.WriteLine($"Received data from input.txt. N = {n}\n\nCalculating...");
9. Stopwatch stopwatch = new Stopwatch();
10. stopwatch.Start();
11. // Initialize dp array
12. int[] dp = new int[n + 1];
13. dp[1] = 0;
14. // Calculate the minimum number of operations for each number i
15. for (int i = 2; i <= n; i++)
16. {
17. dp[i] = dp[i - 1] + 1;
18. if (i % 2 == 0) dp[i] = Math.Min(dp[i], dp[i / 2] + 1);
19. if (i % 3 == 0) dp[i] = Math.Min(dp[i], dp[i / 3] + 1);
20. }
21. stopwatch.Stop();
22. Console.WriteLine($"Result:{dp[n]}");
23. Console.WriteLine($"Execution Time: {stopwatch.Elapsed.TotalMilliseconds} ms");
24. // Write the result to OUTPUT.TXT
25. File.WriteAllText("..//..//..//OUTPUT.TXT", dp[n].ToString());
26. }
27. }

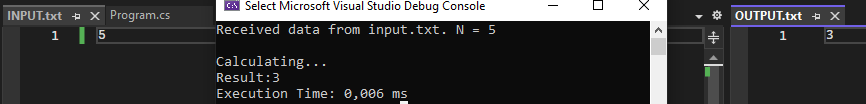
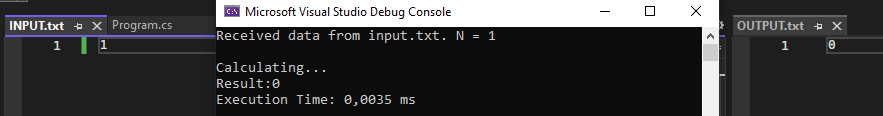
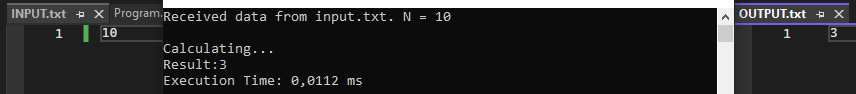
**Опис алгоритму виконання:**

Для розв'язання цієї задачі у було використане динамічне програмування.   
Було створено масив dp розміром n + 1 для зберігання мінімальної кількості операцій, необхідних для зведення кожного числа i до 1. dp[1] присвоюється значення 0, оскільки для зведення 1 до 1 не потрібно виконувати жодних операцій.   
Для кожного наступного числа i обчислювати dp[i] обчислюється як мінімальне з наступних:

* dp[i - 1] + 1 (віднімається 1)
* dp[i / 2] + 1 (ділиться на 2), якщо i парне
* dp[i / 3] + 1 (ділиться на 3), якщо i кратне 3

Врешті-решт, отримуємо dp[n], що дає мінімальну кількість операцій для зменшення n до 1.

**Результати виконання програми:**

Для більшої наочності в програму було додано обчислення часу, що витрачається на виконання обрахунку.  
  
  
1. Input = 5   
Output = 3  
Час виконання = 0,006 мс  
  
  
  
2. Input = 1  
Output = 0  
Час виконання = 0,0035 мс  
  
  
  
3. Input = 10  
Output = 3  
Час виконання = 0,0112 мс  
  
  
4. Input = 106  
Output = 8  
Час виконання = 0,0072 мс  
