

## **Звіт**

### **До лабораторної роботи №4**

По дисципліні «Програмування та алгоритмічні мови»

**Тема: Символи та рядки C#**

**Варіант №6**

**Виконав:** студент гр.

КС-231, Тетієвський В.В.

### **1.Завдання:**

Потрібно ввести одне натуральне (ціле строго додатне) число  $n$  і за ним сформуванати рядок, що містить записані всі підряд натуральні числа від 1 до  $n$ , розділені одинарними пробілами. Наприклад, при  $n = 5$  це повинен бути рядок "1 2 3 4 5". Напишіть чотири версії програми:

1. З використанням лише типу `String`, та додаванням чисел 1, 2, 3, ...,  $n$  у кінець за допомогою оператора `+=` типу `String`;
2. З використанням лише типу `String`, та додаванням чисел  $n$ ,  $n-1$ , ..., 3, 2, 1 (саме в цьому порядку) в початок за допомогою оператора `+` типу `String`;
3. З використанням як `String`, так і `StringBuilder`, та додаванням чисел 1, 2, 3, ...,  $n$  у кінець за допомогою метода `Append` типу `StringBuilder`;
4. З використанням як `String`, так і `StringBuilder`, та додаванням чисел  $n$ ,  $n-1$ , ..., 3, 2, 1

## 2.Текст програм:

1.

```
using System;
using System.Diagnostics;

namespace Da
{
    internal class Program
    {
        static void Main()
        {
            var SB = new Stopwatch();
            int n = int.Parse(Console.ReadLine());
            string s = "";
            SB.Start();
            for (int i = 1; i <= n; i++)
            {
                s += i + " ";
            }
            SB.Stop();
            Console.WriteLine($"{s}\nВитрачено часу:{SB.Elapsed}");
        }
    }
}
```

2.

```
using System;
using System.Diagnostics;

namespace Da
{
    internal class Program
    {
        static void Main()
        {
            var SB = new Stopwatch();
            int n = int.Parse(Console.ReadLine());
            string cur = "";
            SB.Start();
            for (; n >= 1;)
            {
                cur = n + " " + cur;
                n--;
            }
            SB.Stop();
            Console.WriteLine($"{cur}\nЧас виконання:{SB.Elapsed}");
        }
    }
}
```

3.

```
using System;
```

```

using System.Diagnostics;

namespace Da
{
    internal class Program
    {
        static void Main()
        {
            var SB = new Stopwatch();
            int cur = int.Parse(Console.ReadLine());
            StringBuilder sb = new StringBuilder();
            SB.Start();
            for (int i = 1; i <= cur; i++)
            {
                sb.Append(i + " ");
            }
            SB.Stop();
            Console.WriteLine($"{sb}\nВитрачено часу на виконання: {SB.Elapsed}");
        }
    }
}

```

4.

```

using System;
using System.Collections.ObjectModel;
using System.Diagnostics;

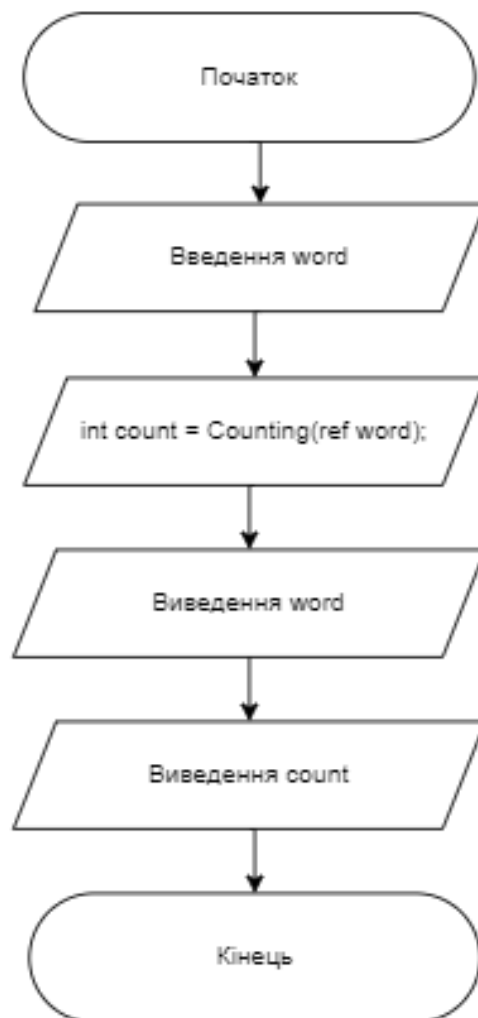
namespace Da
{
    internal class Program
    {
        static void Main()
        {
            var SW = new Stopwatch();
            int n = int.Parse(Console.ReadLine());
            StringBuilder cur = new StringBuilder();
            SW.Start();
            for (; n >= 1;)
            {
                cur.Insert(0, n + " ");
                n--;
            }
            SW.Stop();
            Console.WriteLine($"{cur}\nВитрачено часу на виконання: {SW.Elapsed}");
        }
    }
}

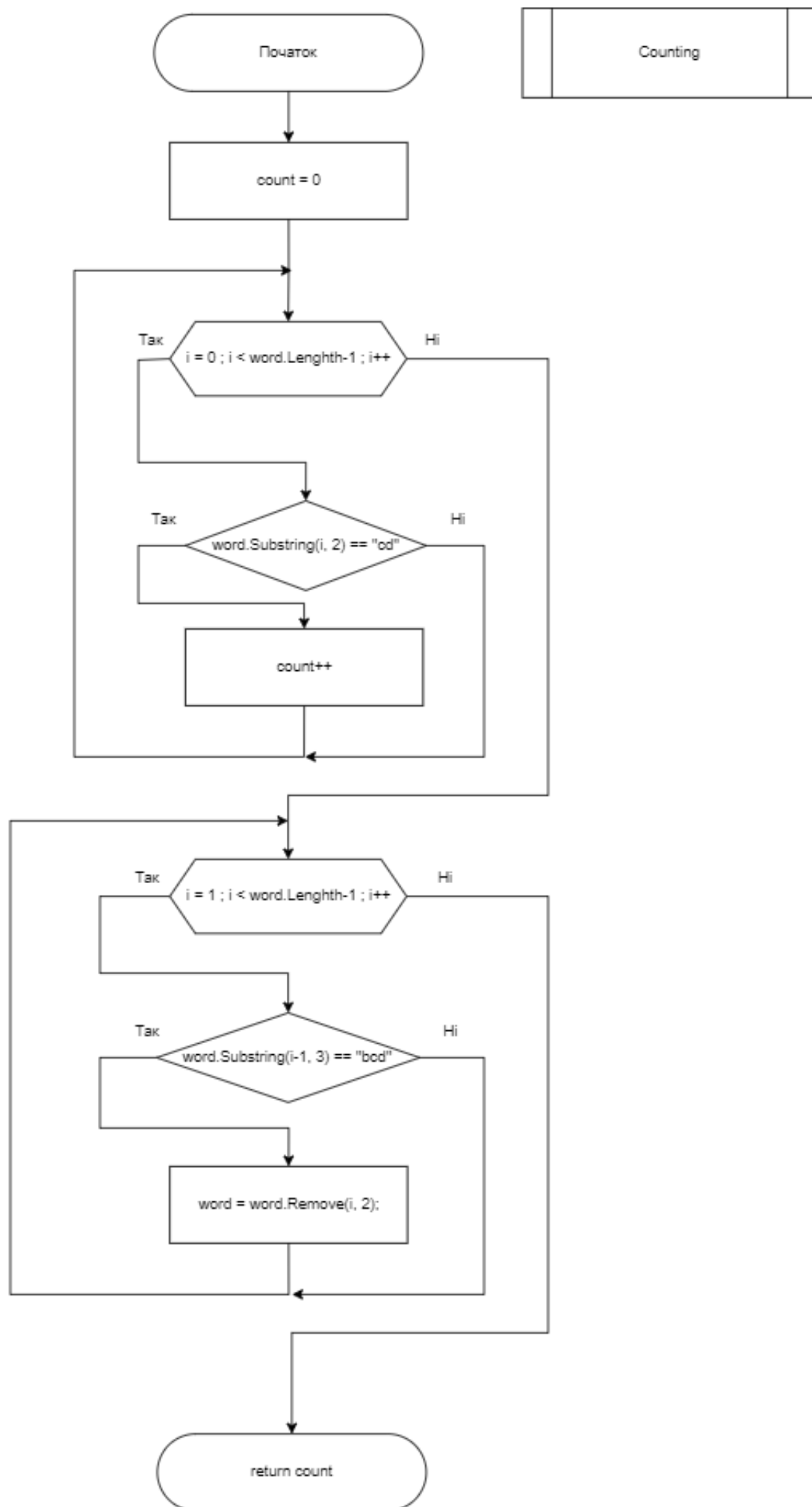
```

### 3. Постановка завдання:

Порахувати кількість послідовностей символів "cd" у рядку і видалити ті з них, перед якими стоїть літера 'b' (видаляти треба лише "cd", а 'b' залишати; наприклад, "abcdcde" повинно перетворитися в "abcde", не в "acde" і не в "abe", водночас, треба виконувати все-таки всі заміни, наприклад, "abcdcdeabcdcde" повинно перетворитися в "abcdeabcde").

Блок схеми:





#### 4.Текст програми для вирішення завдань:

```
using System;
using System.Text;
using System.ComponentModel;

namespace Da
{
    internal class Program
    {
        static void Main()
        {
            Console.OutputEncoding = Encoding.UTF8;
            Console.WriteLine($"6. Введіть рядок:");
            string word = Console.ReadLine();
            int count = Counting(ref word);
            Console.WriteLine($"Результат: {word}");
            Console.WriteLine($"Кількість cd = {count}");
        }
        public static int Counting(ref string word)
        {
            int count = 0;
            for (int i = 0; i < word.Length - 1; i++)
            {
                if (word.Substring(i, 2) == "cd")
                {
                    count++;
                }
            }
            for (int i = 1; i < word.Length - 1; i++)
            {
                if (word.Substring(i - 1, 3) == "bcd")
                {
                    word = word.Remove(i, 2);
                }
            }
            return count;
        }
    }
}
```

#### Опис:

Ми вводим випадкові букви і програма повинна видалити “cd”, якщо перед ним стоїть буква “b”(тобто “bcd”), при цьому залишити “b”.

#### Посилання:

Тут всі задачі: <https://github.com/TheFreinBoy/Laboratory-4.git>

## 5. Приклади вхідних даних та їх результати

	task 1	task 2	task 3	task 4
10000	00:00:00.1829040	00:00:00.1490942	00:00:00.0013989	00:00:00.2141108
20000	00:00:00.7622816	00:00:00.8778996	00:00:00.0023984	00:00:01.8102324
50000	00:00:06.9519886	00:00:07.8169028	00:00:00.0121076	00:00:14.3439961
100000	00:00:34.4070517	00:00:34.2567825	00:00:00.0117137	00:01:02.8815005
200000	00:02:49.7680452	00:03:18.0465104	00:00:00.0248027	00:07:18.4948440

1. У першому завданні я використовував просто string;
2. У другому завданні вже StringBuilder та метод Append;
3. У третьому завданні я використовував знову ж просто string;
4. І в четвертому завданні знову StringBuilder, але вже інший метод Insert;

Як ми бачимо чим більше число тим довше програма його обраховує. Також помітно що завдання зі звичайним String працює швидше ніж зі StringBuilder, але загалом, використання StringBuilder може призвести до більш ефективного використання ресурсів, оскільки цей клас дозволяє ефективно змінювати рядок без створення нових об'єктів під час кожного циклу. Також, я думаю, тут же грає роль потужність комп'ютера.

## 6 Варіант:

```
6. Введіть рядок:abcdcde
Результат: abcde
Кількість cd = 2
```

```
6. Введіть рядок:abcdcdeabcdcde
Результат: abcdeabcde
Кількість cd = 4
```

## 7. Висновок:

Отже, після виконання цієї лабораторної роботи я отримав важливі навички використання рядка String та StringBuilder. Використання цих типів даних дозволяє зберігати та обробляти текстові дані в програмах на C#. Символи використовуються для представлення одного символу, такого як буква чи цифра, тоді як рядки служать для представлення послідовності символів, тобто тексту. Також навчився застосовувати підрядок Substring.