

Міністерство освіти і науки України
ЧЕРКАСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ім. Богдана Хмельницького

Факультет Обчислювальної техніки, інтелектуальних та управляючих систем
Кафедра Програмного забезпечення автоматизованих систем

ЛАБОРАТОРНА РОБОТА № 4

по дисципліні «Програмування та алгоритмічні мови»

Тема: Використання операторів циклу та вибору мови C#

Варіант 6

Виконав: студент гр. КС-231
Киба Д.В.

Черкаси, 2023

1.1 Постановка завдання:

Завдання №1. Потрібно ввести одне натуральне (ціле строго додатне) число n і за ним сформувати 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 кінець за допомогою метода `Append` типу `StringBuilder`;
4. З використанням як `String`, так і `StringBuilder`, та додаванням чисел n , $n-1$, ..., 3, 2, 1 (саме в цьому порядку) в початок за допомогою метода `Insert` типу `StringBuilder`.

1.2 Текст програм для вирішення задач:

1.

```
using System;
using System.Text;
using System.Diagnostics;

namespace Lab4
{
    class Program
    {
        static void Main(string[] args)
        {
            long memoryBefore = GC.GetTotalMemory(true);
            Console.OutputEncoding = UTF8Encoding.UTF8;
            System.Console.WriteLine("Введіть натуральне число n: ");
            // uint n = uint.Parse(Console.ReadLine());
            uint n = 50000;
            string result = "1";
            var watch = Stopwatch.StartNew();
            for (uint i = 2; i ≤ n; i++)
            {
                result += $" {i}";
            }
            System.Console.WriteLine(result);
            watch.Stop();
            long memoryAfter = GC.GetTotalMemory(true);
            long consumedMemory = memoryAfter - memoryBefore;
            System.Console.WriteLine($"Пам'яті спожито: {consumedMemory} байт або
{consumedMemory/1024.0} кілобайт");
            System.Console.WriteLine($"Час виконання програми: {watch.ElapsedMilliseconds}мс");
        }
    }
}
```

```
}  
}
```

2.

```
using System;  
using System.Text;  
using System.Diagnostics;  
  
namespace Lab4  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            long memoryBefore = GC.GetTotalMemory(true);  
            Console.OutputEncoding = UTF8Encoding.UTF8;  
            System.Console.Write("Введіть натуральне число n: ");  
            uint n = uint.Parse(Console.ReadLine());  
            // uint n = 50000;  
            string result = n.ToString();  
  
            var watch = Stopwatch.StartNew();  
            for (uint i = n - 1; i > 0; i--)  
            {  
                result = i.ToString() + " " + result;  
            }  
  
            watch.Stop();  
            System.Console.WriteLine(result);  
            long memoryAfter = GC.GetTotalMemory(true);  
            long consumedMemory = memoryAfter - memoryBefore;  
            System.Console.WriteLine($"Пам'яті спожито: {consumedMemory} байт або  
{consumedMemory / 1024.0} кілобайт");  
            System.Console.WriteLine($"Час виконання програми: {watch.ElapsedMilliseconds}мс");  
        }  
    }  
}
```

3.

```
using System;
using System.Text;
using System.Diagnostics;

namespace Lab4
{
    class Program
    {
        static void Main(string[] args)
        {
            long memoryBefore = GC.GetTotalMemory(true);
            Console.OutputEncoding = UTF8Encoding.UTF8;
            System.Console.Write("Введіть натуральне число n: ");
            uint n = uint.Parse(Console.ReadLine());
            // uint n = 10000;

            StringBuilder result = new StringBuilder();
            result.Append(1);
            var watch = Stopwatch.StartNew();
            for (uint i = 2; i ≤ n; i++)
            {
                result.Append(i.ToString());
                result.Append(" ");
            }

            System.Console.WriteLine(result);
            watch.Stop();
            long memoryAfter = GC.GetTotalMemory(true);
            long consumedMemory = memoryAfter - memoryBefore;
            System.Console.WriteLine();
            System.Console.WriteLine($"Пам'яті спожито: {consumedMemory} байт або
{consumedMemory / 1024.0} кілобайт");
            System.Console.WriteLine($"Час виконання програми: {watch.ElapsedMilliseconds}мс");
        }
    }
}
```

4.

```
using System;
using System.Text;
using System.Diagnostics;

namespace Lab4
{
    class Program
    {
        static void Main(string[] args)
        {
            long memoryBefore = GC.GetTotalMemory(true);
            Console.OutputEncoding = UTF8Encoding.UTF8;
            System.Console.Write("Введіть натуральне число n: ");
            uint n = uint.Parse(Console.ReadLine());
            // uint n = 50;

            StringBuilder result = new StringBuilder(n.ToString());

            var watch = Stopwatch.StartNew();
            for (uint i = n - 1; i > 0; i--)
            {
                result.Insert(0, i.ToString() + " ");
            }

            watch.Stop();
            System.Console.WriteLine(result);
            long memoryAfter = GC.GetTotalMemory(true);
            long consumedMemory = memoryAfter - memoryBefore;
            System.Console.WriteLine();
            System.Console.WriteLine($"Пам'яті спожито: {consumedMemory} байт або
{consumedMemory / 1024.0} кілобайт");
            System.Console.WriteLine($"Час виконання програми: {watch.ElapsedMilliseconds}мс");
        }
    }
}
```

1.3 Текст програми для вирішення завдань мовою Golang:

1.

```
package main

import (
    "fmt"
    "time"
)

func main() {
    var n uint
    fmt.Print("Введіть натуральне число: ")
    fmt.Scan(&n)
    // n = 50000
    result := "1"
    startTime := time.Now()

    for i := uint(2); i ≤ n; i++ {
        result += fmt.Sprintf(" %d", i)
    }
    fmt.Println(result)

    endTime := time.Now()
    elapsedTime := endTime.Sub(startTime).Milliseconds()
    fmt.Printf("Execution time: %dms\n", elapsedTime)
}
```

2.

```
package main

import (
    "fmt"
    "time"
)

func main() {
    var n uint
    fmt.Print("Введіть натуральне число: ")
    fmt.Scan(&n)
    // n = 40000
    result := fmt.Sprintf("%d", n)
```

```

startTime := time.Now()
for i := uint(n - 1); i > 0; i-- {
    result = fmt.Sprintf("%d %s", i, result)
}

fmt.Println(result)

endTime := time.Now()
elapsedTime := endTime.Sub(startTime).Milliseconds()
fmt.Printf("Execution time: %dms\n", elapsedTime)
}

```

3.

```
package main
```

```

import (
    "fmt"
    "strings"
    "time"
)

```

```

func main() {
    var n uint
    fmt.Print("Введіть натуральне число: ")
    fmt.Scan(&n)
    // n = 400

    var builder strings.Builder
    builder.WriteString("1")

    startTime := time.Now()
    for i := uint(2); i ≤ n; i++ {
        builder.WriteString(fmt.Sprintf(" %d", i))
    }
    result := builder.String()
    fmt.Println(result)

    endTime := time.Now()
    elapsedTime := endTime.Sub(startTime).Milliseconds()
    fmt.Printf("Execution time: %dms\n", elapsedTime)
}

```

4.

```
package main

import (
    "fmt"
    "strings"
    "time"
)

func Insert(b *strings.Builder, index int, str string) {
    s := b.String()
    b.Reset()
    b.WriteString(s[:index] + str + s[index:])
}

func main() {
    var n uint
    fmt.Print("Введіть натуральне число n: ")
    fmt.Scan(&n)
    // n = 400

    var builder strings.Builder
    startTime := time.Now()
    for i := uint(n); i > 0; i-- {
        Insert(&builder, 0, fmt.Sprintf("%d ", i))
    }
    result := builder.String()
    fmt.Println(result)

    endTime := time.Now()
    elapsedTime := endTime.Sub(startTime).Milliseconds()
    fmt.Printf("Execution time: %dms\n", elapsedTime)
}
```

1.4 Опис вхідних даних

У програму вводиться ціле невід'ємне число n.

1.5 Посилання на текст програми:

<https://github.com/dimakyba/Lab4>

<https://ideone.com/PgVbyQ> 1

<https://ideone.com/l4UhDp> 2

<https://ideone.com/ZlMUck> 3

<https://ideone.com/NFqwQn> 4

1.6 Заміри швидкості чотирьох версій (C#):

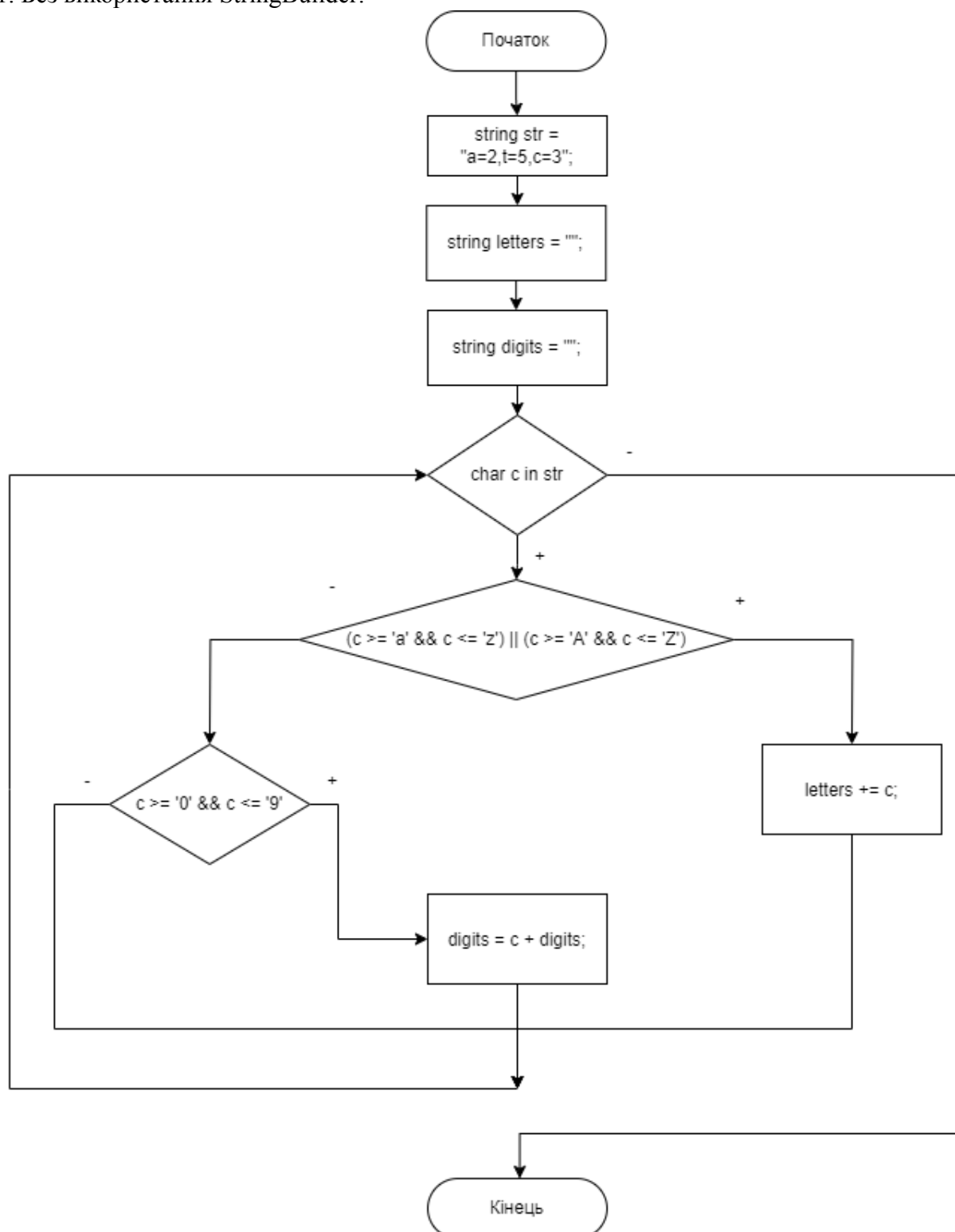
C# (Windows)					
	Task1v1				
	10000	20000	50000	100000	200000
1	99	263	1896	13300	-
2	77	240	2164	-	-
3	75	313	1850	-	-
4	84	243	1851	-	-
5	73	304	1887	-	-
average	81,6	272,6	1929,6	13300	#DIV/0!
	Task1v2				
	10000	20000	50000	100000	200000
1	23	169	1778	13065	-
2	25	136	1724	12848	-
3	27	128	1645	-	-
4	23	135	1720	-	-
5	24	131	1707	-	-
average	24,4	139,8	1714,8	12956,5	#DIV/0!
	Task1v3				
	10000	20000	50000	100000	200000
1	49	119	299	562	1342
2	48	104	276	612	1281
3	56	108	414	576	1226
4	48	102	289	627	1289
5	48	103	279	558	1299
average	49,8	107,2	311,4	587	1287,4
	Task1v4				
	10000	20000	50000	100000	200000
1	81	454	3020	22724	-
2	69	451	3019	-	-
3	68	438	2989	-	-
4	69	436	3016	-	-
5	67	438	2976	-	-
average	70,8	443,4	3004	22724	#DIV/0!

1.1 Постановка завдання

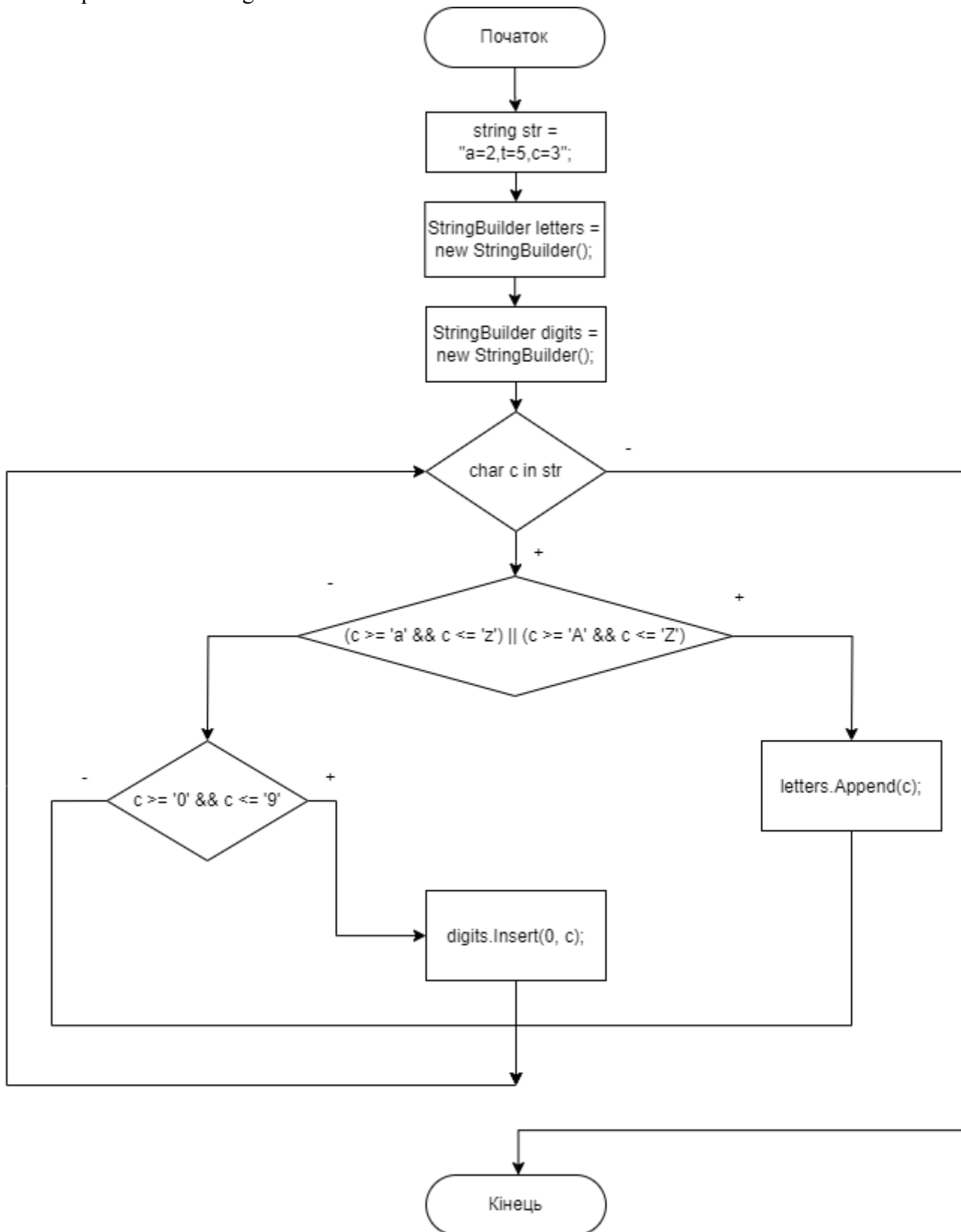
Завдання 2 (Варіант 10). Відомо, що у рядку є латинські (англійські) букви і цифри. Перетворити рядок так, щоб спочатку розміщувалися букви у прямому порядку, а потім цифри у зворотному порядку. Якщо рядок містив також інші символи (не букви й не цифри), їх слід пропустити (видалити). («Букви у прямому порядку» означає, що порядок букв між собою (не відносно решти символів, а між собою) лишається таким, як був. «Цифри у зворотному порядку» означає, що порядок цифр між собою (не відносно решти символів, а між собою) в точності протилежний відносно того, яким був. Наприклад, рядок "a=2,t=5,c=3" слід перетворити у рядок "atc352".)

1.2 Алгоритм вирішення завдання:

1. Без використання StringBuilder:



2. З використанням StringBuilder:



1.3 Текст програм для вирішення завдань:

1. Без використання StringBuilder:

```
using System;

class Program
{
    static void Main(string[] args)
    {
        string str = "a=2,t=5,c=3";

        string letters = "";
        string digits = "";

        foreach (char c in str)
        {
            if ((c ≥ 'a' && c ≤ 'z') || (c ≥ 'A' && c ≤ 'Z'))
            {
                letters += c;
            }
            else if (c ≥ '0' && c ≤ '9')
            {
                digits = c + digits;
            }
        }

        string result = letters + digits;
        Console.WriteLine(result);
    }
}
```

2. З використанням StringBuilder:

```
using System;
using System.Text;

namespace Program
{
    class Program
    {
        static void Main(string[] args)
```

```

{
    string str = "a=2,t=5,c=3";

    StringBuilder letters = new StringBuilder();
    StringBuilder digits = new StringBuilder();

    foreach (char c in str)
    {
        if ((c ≥ 'a' && c ≤ 'z') || (c ≥ 'A' && c ≤ 'Z'))
        {
            letters.Append(c);
        }
        else if (c ≥ '0' && c ≤ '9')
        {
            digits.Insert(0, c);
        }
    }
    string result = letters.ToString() + digits.ToString();
    Console.WriteLine(result);
}
}
}

```

1.4 Текст програм для вирішення завдань мовою Golang:

1. Без використання StringBuilder:

```

package main

import (
    "fmt"
)

func main() {
    str := "a=2,t=5,c=3"

    var letters, digits string

    for _, c := range str {
        if c ≥ 'a' && c ≤ 'z' || c ≥ 'A' && c ≤ 'Z' {
            letters += string(c)
        } else if c ≥ '0' && c ≤ '9' {

```

```

        digits = string(c) + digits
    }
}

result := letters + digits
fmt.Println(result)
}

```

2.3 використанням StringBuilder:

```

package main

import (
    "fmt"
    "strings"
)

func main() {
    str := "a=2,t=5,c=3"

    var letters strings.Builder
    var digits strings.Builder

    for _, c := range str {
        if c ≥ 'a' && c ≤ 'z' || c ≥ 'A' && c ≤ 'Z' {
            letters.WriteRune(c)
        } else if c ≥ '0' && c ≤ '9' {
            Insert(&digits, 0, fmt.Sprintf("%c", c))
        }
    }

    result := letters.String() + digits.String()
    fmt.Println(result)
}

func Insert(b *strings.Builder, index int, str string) {
    s := b.String()
    b.Reset()
    b.WriteString(s[:index] + str + s[index:])
}

```

1.5 Посилання на текст програми:

<https://ideone.com/pFrEzH>

<https://ideone.com/PzVLlJ>