



НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ  
ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки

ЗВІТ ДО ПРАКТИЧНОЇ РОБОТИ  
з дисципліни  
«Технології програмування на C/Embedded»  
на тему «Строки»

Перевірив:  
Каплунов Артем  
Володимирович

Виконав:  
студент групи ІК-34  
Могильний Микола

## Строки

**Мета роботи:** отримати початкові навички роботи з С.

**Завдання:** Користувач вводить строку з клавіатури. Додаток оброблює її згідно індивідуального завдання (див. нижче) та виводить отриманий результат на екран. Якщо завдання цілком вирішується за допомогою однієї з бібліотечних функцій, то її використання заборонене (наприклад, "виверт строки" можна вирішити за допомогою функції `ReverseString`). Додаток повинен працювати з символами англійської мови, числами і т.і. (перша половина таблиці ASCII), обробку кирилиці не виконувати.

**Послідовність виконання:**

- 1) Створити додаток виконуючий дії згідно вашого завдання.

## Хід роботи

1. Створити додаток виконуючий дії згідно вашого завдання.

14. Подвоєння кожного симола в строці.

Посилання на GitHub: [Посилання](#)

Посилання на мій коміт на GitHub: [Посилання](#)

## Код для розрахунку строки

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

#define MAX_INPUT_SIZE 101

char* double_characters(const char* input_string) {
    // 1. Calculate the length of the input string
    size_t input_length = strlen(input_string);

    // 2. Calculate the required length for the new string:
    size_t new_length = (input_length * 2) + 1;

    // 3. Allocate memory for the new string dynamically
    char* new_string = (char*)malloc(new_length * sizeof(char));

    // Check if memory allocation failed
    if (new_string == NULL) {
        perror("Memory allocation error");
        return NULL;
    }

    // 4. Iterate through the input string and populate the new string
    size_t i = 0;
    size_t j = 0;
```

```

while (i < input_length) {
    char current_char = input_string[i];
    new_string[j] = current_char;
    new_string[j + 1] = current_char;
    i++;
    j += 2;
}

// 5. Terminate the new string with the null character '\0'
new_string[new_length - 1] = '\0';

// 6. Return the new string
return new_string;
}

int main() {
    char input[MAX_INPUT_SIZE];
    char* result_string = NULL;

    printf("Enter a string (max %d characters): ", MAX_INPUT_SIZE - 1);

    // Read the string from the keyboard safely using fgets.
    if (fgets(input, MAX_INPUT_SIZE, stdin) == NULL) {
        fprintf(stderr, "Error reading input.\n");
        return 1;
    }

    // fgets includes the newline character '\n'.
    // Remove it if it exists.
    size_t len = strlen(input);
    if (len > 0 && input[len - 1] == '\n') {
        input[len - 1] = '\0';
    }

    // Process the string
    result_string = double_characters(input);

    // Output the result
    if (result_string != NULL) {
        printf("\nOriginal string: %s\n", input);
        printf("Doubled string: %s\n", result_string);
        free(result_string);
    } else {
        fprintf(stderr, "\nFailed to process the string.\n");
        return 1;
    }

    return 0;
}

```

## Результат виконання

```
Enter a string (max 100 characters):Glory to Ukraine
```

```
Original string: Glory to Ukraine
```

```
Doubled string: GGllloorrryy tttoo UUKkrrraaiinneee
```

## **Висновок**

У ході виконання лабораторної роботи було успішно засвоєно подвоєння кожного символу у введеній строці — досягнута шляхом створення функції `double_characters`, яка обходить заборону використання готових бібліотечних функцій. Робота додала розуміння механізмів обробки ASCII-строк та принципів управління пам'яттю в мові C.