

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 |

Київ 2025

**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.

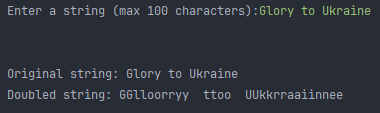
Посилання на GitHub: [Посилання](https://github.com/quantum200?tab=repositories)

Посилання на мій коміт на GitHub: [Посилання](https://github.com/quantum200/C-Embedden/tree/aae0141ddd3651bcc7d10ca1b4b77a4b913cec11/%D0%9A%D0%9F3)

**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



**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.