

Лабораторна робота №12. Взаємодія з користувачем шляхом механізму вводу/виведення

1. Вимоги

1.1 Розробник

- Журавльов Ярослав Юрійович;
- студент групи КІТ-120а;
- 15-лют-2021

1.2 Загальне завдання

Розробити функції, які використовують механізми вводу/виведення для демонстрації роботи програми.

1.3 Індивідуальне завдання

Продемонструвати взаємодію з користувачем шляхом використання `printf()`, `scanf()`, `gets()`, `puts()`, `getc()`, `putc()`, `read()`, `write()`.

2. Опис програми

2.1 Функціональне призначення

Програма приймає текст за допомогою однієї з функцій вводу, визначає числа у тексті та виводить ці числа за допомогою однієї з функцій виведення.

2.2 Опис логічної структури програми

Для вводу тексту створюємо змінну `char *input`, яка буде містити у собі ввід. За допомогою функцій, які використовують різні функції взаємодії запитуємо у користувача текст та виводимо числа у тексті.

Функція виведення за допомогою printf.

```
void print_numbers_by_printf
```

Призначення: приймає числа та виводить у консоль

Схема алгоритму функції подана на рис. 1.

Опис роботи: функція перебирає кожне число *i* та виводить його у консоль.

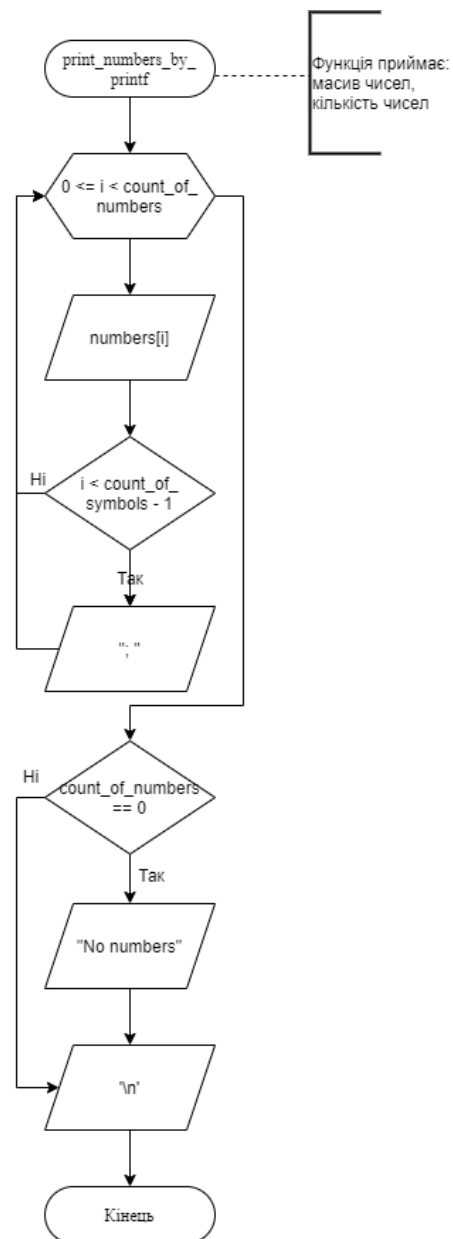


Рисунок 1 — Схема алгоритму функції *print_numbers_by_printf*

Функція створення строки виводу.

```
void create_output_string
```

Призначення: приймає числа та створює з ними строку для виводу

Схема алгоритму функції подана на рис. 2.

Опис роботи: функція перебирає усі елементи масиву чисел і записує числа у строку.



Рисунок 2 — Схема алгоритму функції `create_output_string`

Функція виведення за допомогою puts.

```
void print_numbers_by_puts
```

Призначення: приймає числа та виводить у консоль

Схема алгоритму функції подана на рис. 3.

Опис роботи: функція створює строку для виводу за допомогою функції *create_output_string* і виводить її у консоль.

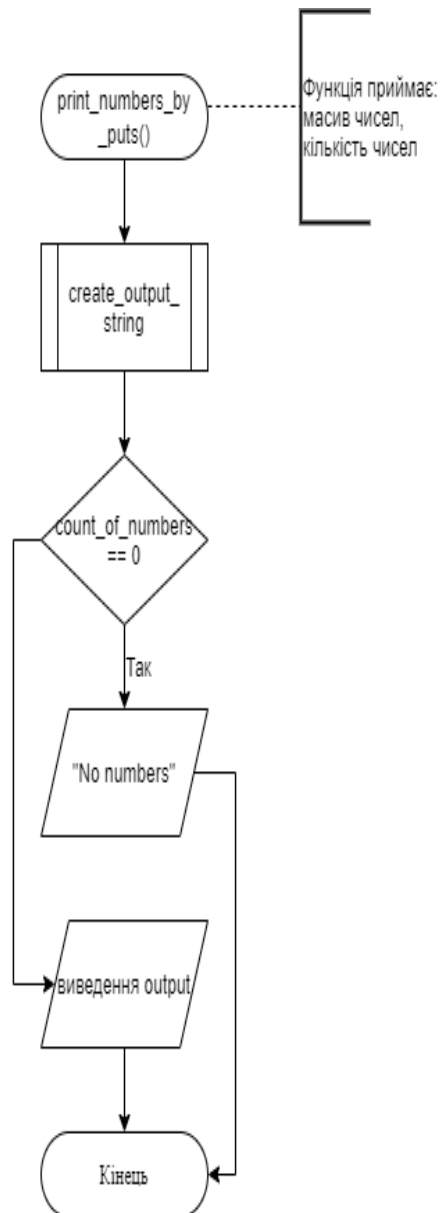


Рисунок 3 — Схема алгоритму функції *print_numbers_by_puts*

Функція введення/виведення за допомогою `getc` та `putc`.

```
void use_getc_and_putc
```

Призначення: зчитує текст з консолі, визначає у ньому числа та виводить у консоль

Схема алгоритму функції подана на рис. 4.

Опис роботи: функція зчитує дані з консолі за допомогою функції `getc`, знаходить у тексті числа та виводить їх у консоль за допомогою функції `putc`.

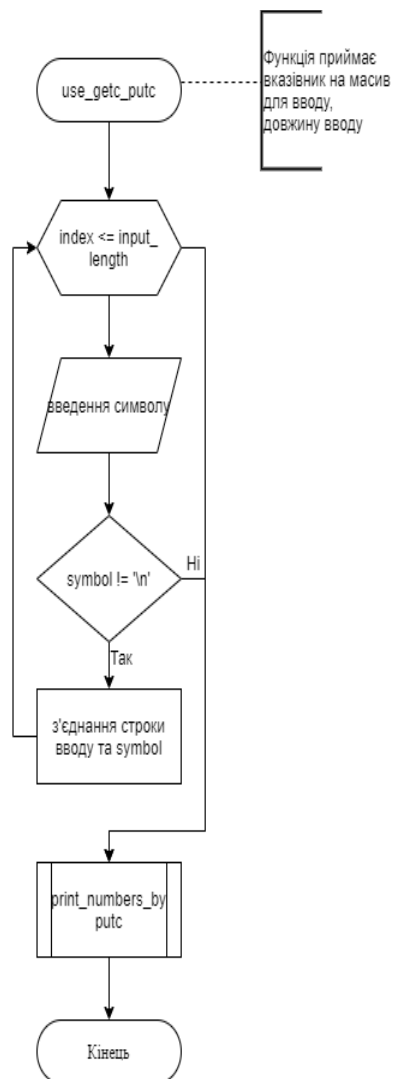


Рисунок 4 — Схема алгоритму функції `use_getc_and_putc`

Функція виведення за допомогою puts.

```
void print_numbers_by_putc
```

Призначення: виводить у консоль числа.

Схема алгоритму функції подана на рис. 5.

Опис роботи: функція створює строку для виводу за допомогою функції *create_output_string* і виводить її у консоль.

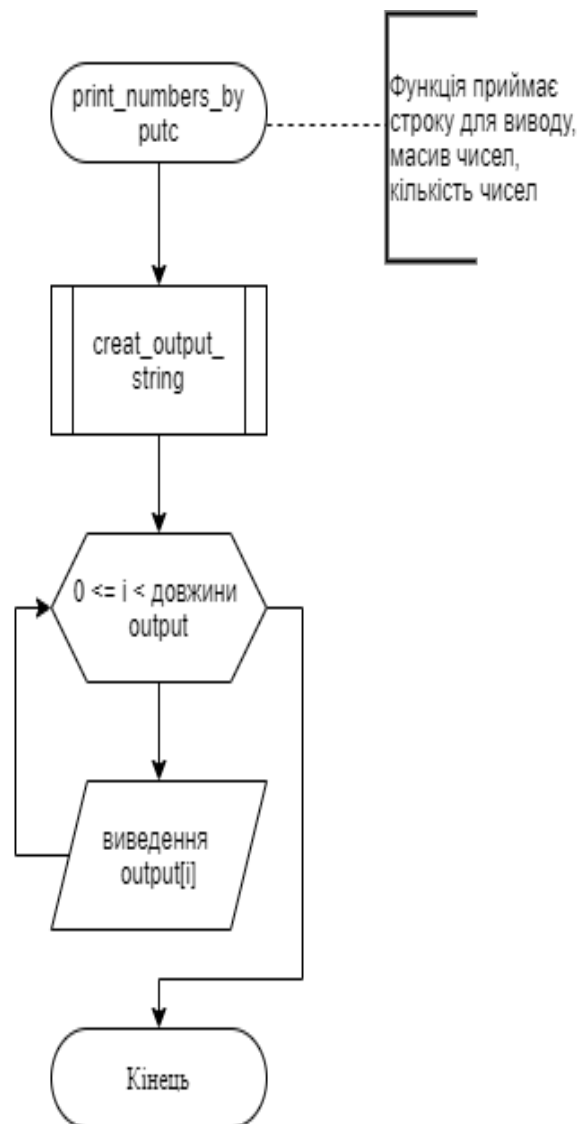


Рисунок 5 — Схема алгоритму функції `print_numbers_by_putc`

Функція виведення за допомогою puts.

```
void print_numbers_by_write
```

Призначення: виводить у консоль числа.

Схема алгоритму функції подана на рис. 6.

Опис роботи: функція створює строку для виводу за допомогою функції `create_output_string` і виводить її у консоль.

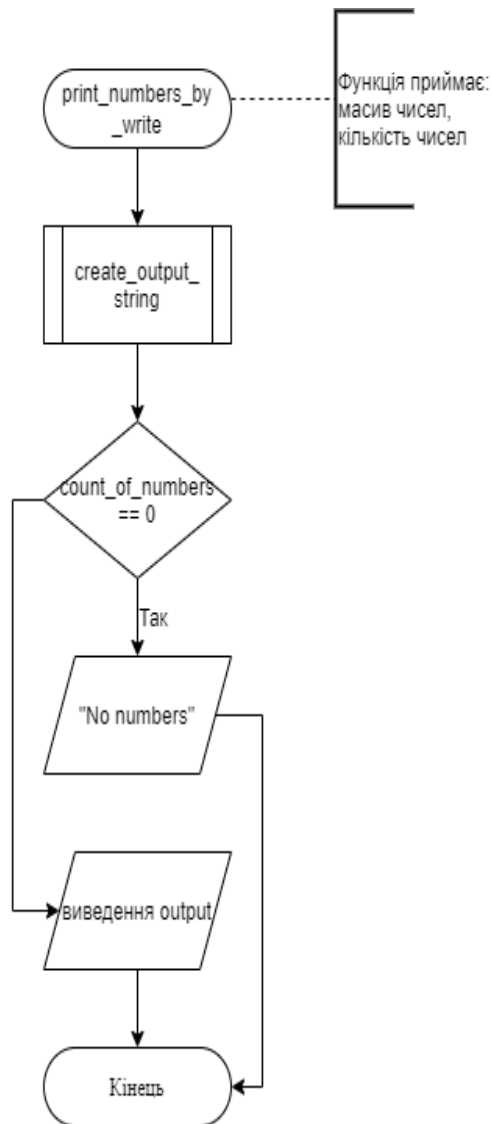


Рисунок 6 — Схема алгоритму функції `print_numbers_by_write`

Основна функція

int main

Призначення: головна функція

Схема алгоритму функції подана на рис. 7.

Опис роботи: по чергово викликає функції роботи з різними функціями вводу/виводу.

Повертає функція код повернення програми (0).

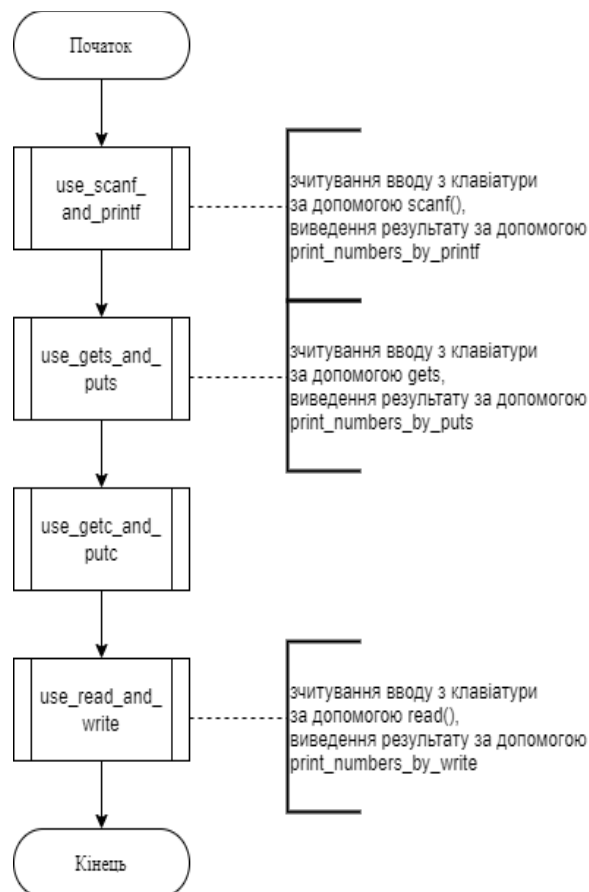


Рисунок 7 — Схема алгоритму функції `main`

Структура проекту

```
└─ lab12
    ├── Doxyfile
    ├── Makefile
    ├── doc
    │   ├── lab07.odt
    │   └─ lab07.pdf
    └─ src
        ├── main.c
        ├── lib.c
        └─ lib.h
```

2.3 Важливі фрагменти програми

Підключення заголовочного `stdio.h` для роботи з користувачем шляхом застосування механізму вводу/виведення.

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

Прийняття вводу за допомогою різних функцій

```
scanf("%255s", input);
fgets(input, 255, stdin);
while ((symbol = (char)getc(stdin)) != '\n' && index <=
input_length)
read(fields, input, nbytes);
```

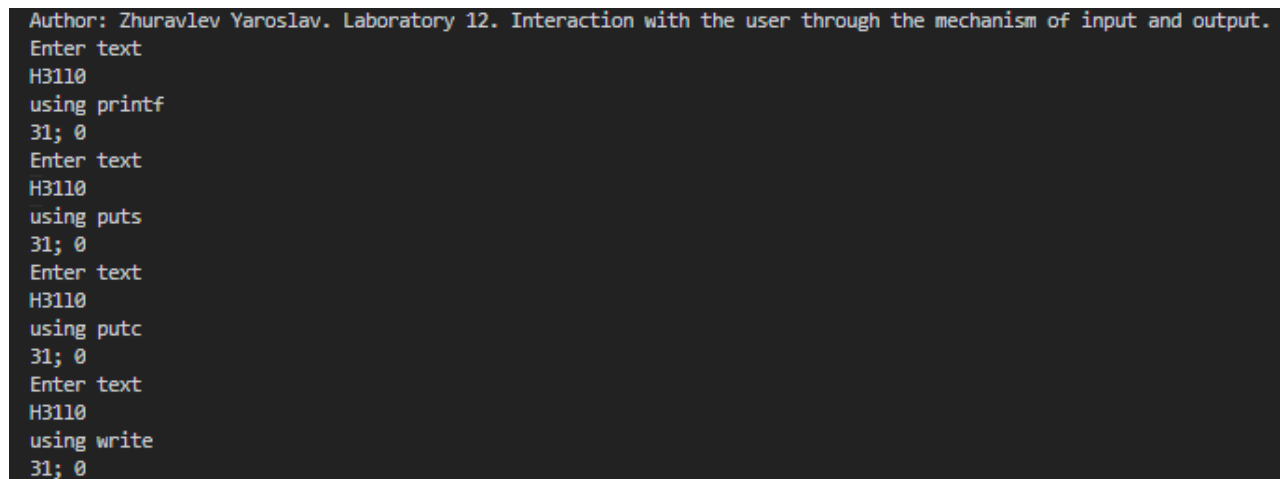
Виведення у консоль за допомогою різних функцій

```
for (size_t i = 0; i < count_of_numbers; i++)
{
    printf("%ld", *(numbers + i));
    if (i < count_of_numbers - 1)
        printf("%s", "; ");
}
puts(output);
for (size_t i = 0; i < strlen(output); i++)
{
    putchar(output[i], stdout);
}
write(fields, output, nbytes);
```

3. Варіанти використання

Для демонстрації результатів використовується виведення результату роботи за допомогою функції виведення.

Програма запитує текст та виводить числа у ньому, результат на рис. 8:



```
Author: Zhuravlev Yaroslav. Laboratory 12. Interaction with the user through the mechanism of input and output.
Enter text
H3110
using printf
31; 0
Enter text
H3110
using puts
31; 0
Enter text
H3110
using putchar
31; 0
Enter text
H3110
using write
31; 0
```

Висновки

При виконанні даної лабораторної роботи було набуто практичного досвіду роботи із механізмом вводу/виводу.