

Міністерство освіти і науки України  
Національний технічний університет  
«Харківський політехнічний інститут»  
Кафедра «Обчислювальна техніка та програмування»

### **ЗВІТ**

Про виконання лабораторної роботи № 7  
«Функції»

Керівник: викладач  
Бульба С. С.

Виконавець: студент гр. КІТ-120в  
Бабенко А. П.

Харків 2020

# Лабораторна робота №7. Функції

## 1. Вимоги

### 1.1 Розробник

- Бабенко Антон Павлович;
- Студент групи КІТ-120в;
- 22 листопада 2020;

### 1.2 Загальне завдання

реалізувати функцію, що визначає, скільки серед заданої послідовності чисел таких пар, у котрих перше число менше наступного, використовуючи функцію з варіативною кількістю аргументів.

### 1.3 Індивідуальне завдання

Переробити програми, що були розроблені під час виконання лабораторних робіт з тем “Масиви” та “Цикли” таким чином, щоб використовувалися функції для обчислення результату.

Функції повинні задовольняти основну їх причетність - уникати дублювання коду. Тому, для демонстрації роботи, ваша програма (функція *main()*) повинна викликати декілька раз розроблену функцію з різними вхідними даними.

## 2 Виконання роботи

2.1 Створення директорії lab07. Зображено на рис.1.

```
anton@anton-X55VD:~/dev/Programming-repo$ mkdir lab07
```

Рисунок 1 - створення директорії

2.2 Створення файлу з вихідним кодом. Зображено на рис.2.

```
anton@anton-X55VD:~/dev/Programming-repo/lab07$ mkdir src
anton@anton-X55VD:~/dev/Programming-repo/lab07$ cd src/
anton@anton-X55VD:~/dev/Programming-repo/lab07/src$ touch task1.c
```

Рисунок 2 - створення файлу з кодом

2.3 Запуск редактора коду. Зображено на рис.3.

```
anton@anton-X55VD:~/dev/Programming-repo/lab07$ kate task1.c
```

Рисунок 3 - запуск редактора

2.4 Написання коду і коментарів до нього. Зображено на рис.4.

```
1 | #include <stdarg.h>
2
3 | int func(int nums, ...)
4 | {
5 |     int count = 0; // Переменная для подсчёта количества пар
6 |     int arr[nums]; // Массив для записи чисел, переданных в функцию
7 |     va_list args; // Объявление списка аргументов
8 |     va_start(args, nums); // Начало работы со списком аргументов
9 |     for (int i = 0; i < nums; i++) {
10 |         arr[i] = va_arg(args, int); // Запись чисел переданных в функцию в массив
11 |     }
12 |     for (int i = 0; i < nums; i++) {
13 |         if (arr[i] < arr[i+1]) { // Сравнение первого и следующего числа из массива
14 |             count++;           // Увеличение количества пар на 1
15 |         }
16 |     }
17 |     va_end(args); // Конец работы со списком аргументов
18 |     return count; // Возвращение результата
19 | }
20
21 | int main()
22 | {
23 |     func(5, 1, 2, 3, 5, 6); // Вызов функции
24 |     return 0;
25 | }
```

Рисунок 4 - код програми

2.5 Компіляція проекту за допомогою команди “make clean prep compile”. Зображено на рис.5.

```
anton@anton-X55VD:~/dev/Programming-repo/lab07$ make clean prep compile
rm -rf dist
mkdir dist
gcc -std=gnu11 -g -Wall -Wextra -Wformat-security -Wfloat-equal -Wshadow -Wconversion -Wlogical-not-parentheses -Wnull-dereference -I./src src/task1.c -o ./dist/task1.bin
```

Рисунок 5 - компіляція проекту

2.6 Відкрито у відлагоднику nemiver виконуючий файл task1.bin. Зображено на рис.6.

The screenshot shows the nemiver debugger interface. The top pane displays C code for a function `func` that takes a variable number of arguments. The code includes comments in Russian. The bottom pane shows the 'Локальные переменные' (Local Variables) window with the following data:

Имя	Значение	Тип
count	0	int
arr	[1]	int [1]
args	[1]	va_list
nums	0	int

At the bottom, there are tabs for 'Терминал цели', 'Контекст', 'Точки останова', 'Регистры', 'Память', and 'Монитор выражений'.

Рисунок 6 - файл у відлагоднику

2.7 Ставимо точку зупину, проходимо по файлу і бачимо зміну значення `count` в момент проходження по циклу. Зображено на рис.7.

This is a close-up of the 'Локальные переменные' window. A red arrow points to the value of the `count` variable, which has changed from 0 to 3. The other variables shown are `arr` (value [1], type int [1]), `args` (value [1], type va\_list), `i` (value 4, type int), and `nums` (value 5, type int).

Имя	Значение	Тип
count	3	int
arr	[1]	int [1]
args	[1]	va_list
i	4	int
nums	5	int

Рисунок 7 - файл у відлагоднику

2.8 Створення блок-схеми програми. Зображення блок-схеми на рис.8.

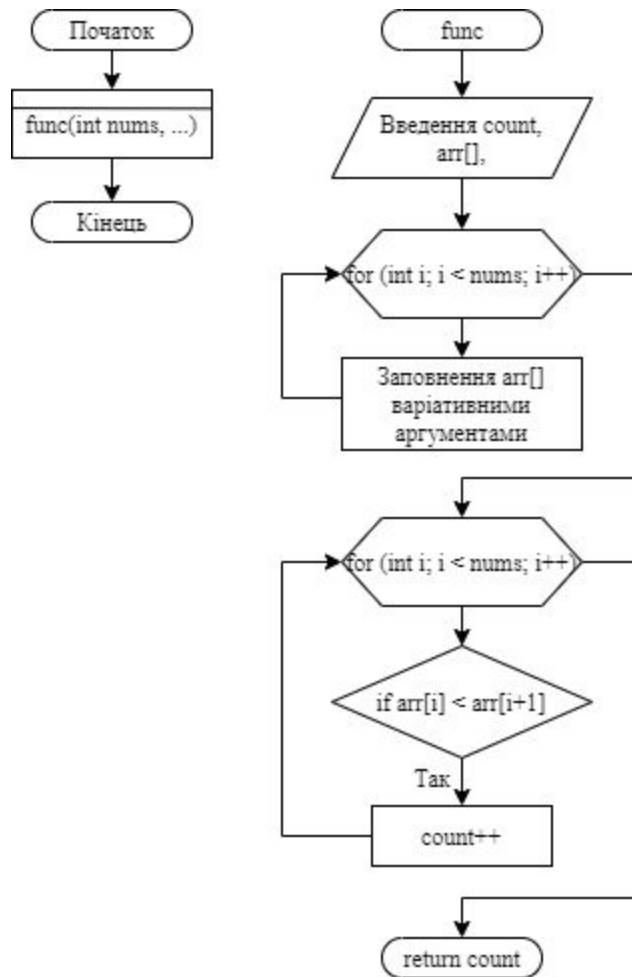


Рисунок 8 - блок-схема

2.9 Перепишемо код програми для обчислення найбільшого спільного дільника двох заданих чисел з використанням функцій, при чому для одного із них використовуються генератор псевдовипадкових чисел. Зображено на рис.9.

```

int nod(int, int);
int get_rand();

int main()
{
    srand(time(0));

    printf("%d\n", nod(6, get_rand()));
    return 0;
}

int nod(int a, int b)
{
    int res; // Объявляем переменную для записи результата
    int i = 1;
    if (a < b) // Запись младшего числа в переменную b
    {
        int x = a;
        a = b;
        b = x;
    }
    for (i = 1; i <= b; i++) // Цикл прохождения от 1 до младшего числа
    {
        if ((a % i) == 0 && (b % i) == 0) // Если остаток от деления двух чисел == 0, записываем его в переменную
        {
            res = i;
        }
    }
    return res;
}

int get_rand() { // Функция получения псевдослучайного числа
    return rand() % 10 + 1;
}

```

Рисунок 9 - код програми

2.10 Створення блок-схеми програми. Зображення блок-схеми на рис.10.

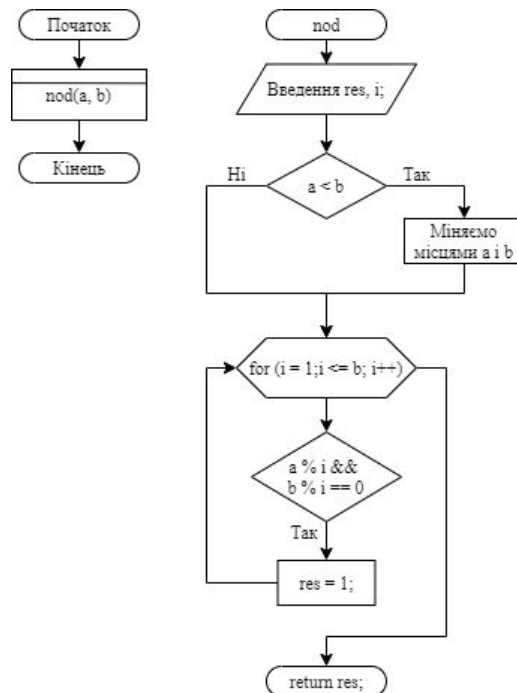


Рисунок 10 - блок-схема

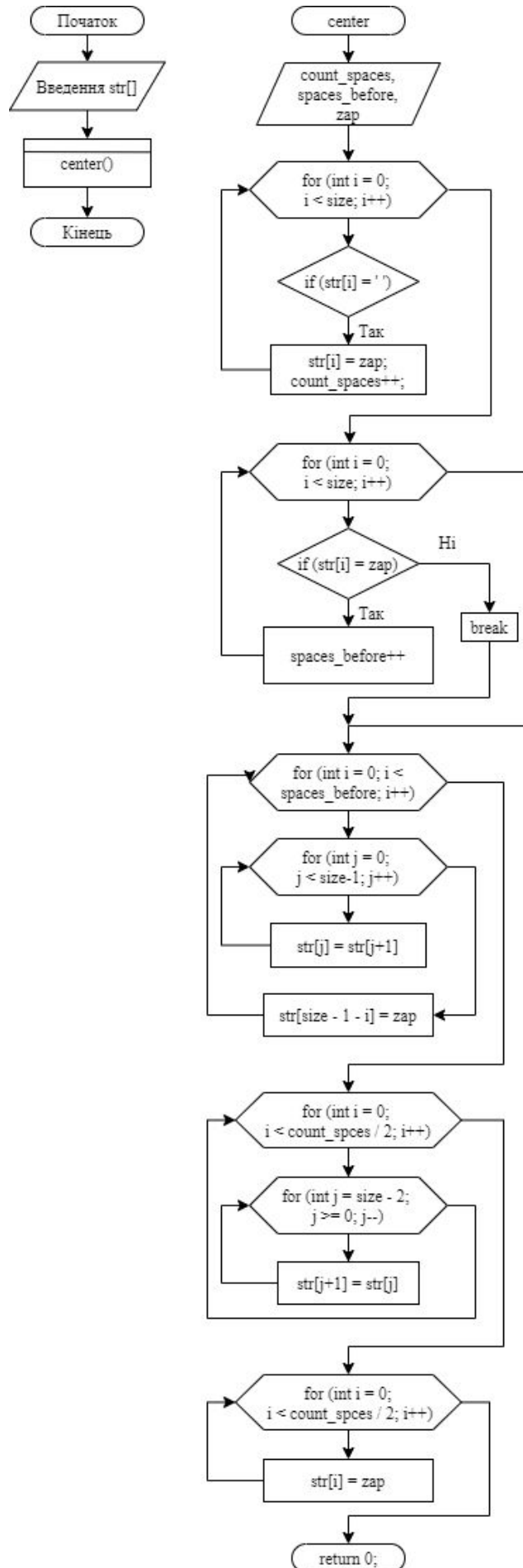
2.11 Перепишемо код програми для центрування слова в масиві з використанням функцій. Зображено на рис.11.

```
char center(char str[]);
int main()
{
    char str[] = "Ivanov    ";
    center(str);
    return 0;
}

char center(char str[]) {
    char zap = ' '; // Задаем заполнитель
    int count_spaces = 0; // Задаем переменные для записи количества пробелов, и пробелов перед словом
    int spaces_before = 0;
    int size = 0;
    for (int i = 0; str[i] != '\0'; i++) { // Вычисляем размер заданного массива
        size += 1;
    }
    for (int i = 0; i < size; i++) { // Меняем пробелы на заполнитель и считаем кол-во пробелов
        if (str[i] == ' ')
        {
            str[i] = zap;
            count_spaces++;
        }
    }
    for (int i = 0; i < size; i++) { // Считаем кол-во пробелов перед символами
        if (str[i] == zap) {
            spaces_before++;
        }
        else {
            break;
        }
    }
    for (int i = 0; i < spaces_before; i++) { // Сдвигаем слово влево, если пробелы перед словом
        for (int j = 0; j < size - 1; j++) {
            str[j] = str[j + 1];
        }
        str[size - 1 - i] = zap;
    }
    for (int i = 0; i < count_spaces / 2; i++) { // Сдвигаем слово на количество пробелов делённое на 2
        for (int j = size - 2; j >= 0; j--) {
            str[j + 1] = str[j];
        }
    }
    for (int i = 0; i < count_spaces / 2; i++) { // Заменяем на заполнитель первые символы кол-во которых равно количеству пробелов пополам
        str[i] = zap;
    }
    return 0;
}
```

Рисунок 11 - код програми

2.12 Створення блок-схеми програми. Зображення блок-схеми на рис.12.





## Рисунок 12 - блок-схема

2.13 Зафіксовано зміни за допомогою команди “git commit” та завантаження до репозиторію за допомогою команди “git push”. Зображено на рис.13, 14.

```
anton@anton-X55VD:~/dev/Programming-repo$ git commit -m "Added lab07"  
[main 051302e] Added lab07
```

## Рисунок 13 - комміт

```
anton@anton-X55VD:~/dev/Programming-repo$ git push  
Перечисление объектов: 19, готово.  
Подсчет объектов: 100% (19/19), готово.  
При сжатии изменений используется до 2 потоков  
Сжатие объектов: 100% (18/18), готово.  
Запись объектов: 100% (18/18), 6.42 KiB | 3.21 MiB/s, готово.  
Total 18 (delta 2), reused 0 (delta 0), pack-reused 0  
remote: Resolving deltas: 100% (2/2), completed with 1 local object.  
To https://github.com/Dewwine/Programming-repo.git  
2c8868a..051302e  main -> main
```

## Рисунок 14 - завантаження змін

## Висновки

При виконанні даної лабораторної роботи було набуто навичок розробки програм з використанням функцій і розроблено 12 програм, а також створено програму, яка відцентровує слово посеред масиву символів та програму, котра шукає найбільший спільний дільник.