

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «Обчислювальної техніки та програмування»

Звіт з лабораторної роботи №7

Тема: «Функції»

Виконав:

ст. гр. КІТ-120в Львов Артем

Перевірив:

Бульба С.С.

Харків 2020

Лабораторна робота №7. Функції

Розробник:

Львов Артем Сергійович

Студент групи КІТ-120В

22.12.2020

Загальне завдання:

Написання та реалізація програм у функціях

Індивідуальне завдання:

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

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

Слід звернути увагу: параметри одного з викликів функції повинно бути згенеровано за допомогою генератора псевдо-випадкових чисел *rand()*.

Функціональне призначення:

Програма демонструє роботу трьох функцій:

1 – функція *primeum()*, яка отримує ціле число, та повертає результат, чи є воно простим, чи ні;

2 – функція *nickname()*, яка отримує масив символів або строку, та перетворює її на нікнейм, замінюючи певні символи на задані;

3 – функція *pairs()*, яка отримує послідовність чисел (з першим числом, яке визначає кількість подальших аргументів), та повертає кількість серед цієї послідовності пар чисел у яких перше число менше за наступне.

Виконання роботи:

1.Створення репозиторію:

Створюю новий репозиторій, клоную в нього репозиторій однієї з попередніх лабораторних робіт, та клоную у систему командою *git clone*

2.Написання коду:

Пишу код програми у файлі Lab_07/src/main.c

```
1 #include <stdarg.h>
2
3 int primenum (int src);           /*prototype of "primenum()" function*/
4 void nickname (char src[]);      /*prototype of "nickname()" function*/
5 int pairs (int cnt, ...);        /*prototype of "pairs()" function*/
6
7 int main()
8 {
9     int test_1 = primenum(rand() % 100); /*test of "primenum()" function*/
10    char test_2[] = {"aois AOIS"};
11    nickname(test_2);               /*test of "nickname()" function*/
12    int test_3 = pairs (4, 2, 4, 6, 8); /*test of "pairs()" function*/
13    return 0;
14 }
```

Рисунок 2.1 – фрагмент коду функції main() та прототипи ін. функцій

```
50 int primenum (int src)
51 {
52     int res = 0;                  /*defining variable for GF*/
53     for (int i = 1; i < src; i++){ /*cycle to calculate the GF*/
54         if (src % i == 0){
55             res = i;
56         }
57     }
58     if (res == 1){
59         return 1;                /*returning result if the number is prime*/
60     }
61     else{
62         return 0;                /*returning result if the number is not prime*/
63     }
64 }
```

Рисунок 2.2 – фрагмент коду функції primenum()

```
66 void nickname (char src[])
67 {
68     for (int i = 0; i <= sizeof(src); i++){ /*cycle for changing defined characters*/
69         if (src[i] == 'a' || src[i] == 'A'){
70             src[i] = '@';
71         }
72         else if (src[i] == 'o' || src[i] == 'O'){
73             src[i] = '0';
74         }
75         else if (src[i] == 'i' || src[i] == 'I'){
76             src[i] = '1';
77         }
78         else if (src[i] == 's' || src[i] == 'S'){
79             src[i] = '$';
80         }
81     }
82 }
```

Рисунок 2.3 – фрагмент коду функції nickname()

```
84 int pairs (const int cnt, ...)
85 {
86     int getres [cnt];             /*defining array for given arguments*/
87     int result = 0;              /*defining variable for resulting*/
88     va_list list;
89     va_start (list, cnt);
90     for (int i = 0; i <= cnt; i++){ /*cycle to write arguments into the array*/
91         getres [i] = va_arg (list, int);
92     }
93     va_end (list);
94     for (int i = 1; i <= cnt; i++){ /*cycle to calculate number of pairs*/
95         if (getres [i] < getres [i + 1]){
96             result ++;
97         }
98     }
99     if (getres [0] > getres [1]){ /*check to fix result if the first arg is bigger than count*/
100         result = result - 1;
101     }
102     return result;               /*returning result*/
103 }
```

Рисунок 2.4 – фрагмент коду функції pairs()

2.1. Компіляція проекту:

Компілюю проект за допомогою команди `make clean prep compile`:

```
o71@071:~/Lvov_Artem/Lab_07$ 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 src/main.c -o ./dist/main.bin
src/main.c: In function 'main':
src/main.c:9:24: warning: implicit declaration of function 'rand' [-Wimplicit-function-declaration]
   9 |     int test_1 = primenum(rand() % 100);
     |                        ^~~~~~
src/main.c:12:6: warning: unused variable 'test_3' [-Wunused-variable]
   12 |     int test_3 = pairs(4, 2, 4, 6, 8);
     |     ^~~~~~
src/main.c:9:6: warning: unused variable 'test_1' [-Wunused-variable]
   9 |     int test_1 = primenum(rand() % 100);
     |     ^~~~~~
src/main.c: In function 'nickname':
src/main.c:68:23: warning: 'sizeof' on array function parameter 'src' will return size of 'char *' [-Wsizeof-array-argument]
   68 |     int arr_size = sizeof(src);
     |                      ^~~~~~
src/main.c:66:21: note: declared here
   66 | void nickname(char src[])
src/main.c:66:21: note: declared here
o71@071:~/Lvov_Artem/Lab_07$
```

Рисунок 2.1 – компіляція проекту

2.2. Перевірка за допомогою nemiver:

Перевіряю роботу проекту командою `nemiver ./dist/main.bin`

The screenshot shows the nemiver debugger interface. The top pane displays the source code of `main.c` with line numbers 1 through 15. The code includes `<stdarg.h>`, defines `primenum`, `nickname`, and `pairs` functions, and contains a `main` function that tests these functions. The bottom pane shows a table of local variables for the current frame (Frame 0).

Threa	F	Variable	Value	Type
1	0	Local Variables		
		test_1	1	int
		test_2	[10]	char[10]
		0	64 '@'	char
		1	48 '0'	char
		2	49 '1'	char
		3	36 'S'	char
		4	32 ''	char
		5	64 '@'	char
		6	48 '0'	char
		7	49 '1'	char
		8	36 'S'	char
		9	0 '\0'	char
		test_3	3	int
		Function Arguments		

The bottom of the window has tabs for Target Terminal, Context, Breakpoints, Registers, Memory, and Expression Monitor.

Рисунок 2.2 – вікно nemiver

3. Завантаження проекту на GitHub:

```
o71@071:~$ cd Lvov_Artem/Lab_07
o71@071:~/Lvov_Artem/Lab_07$ git add .
o71@071:~/Lvov_Artem/Lab_07$ git commit -m"all"
[master cd084a9] all
 2 files changed, 121 insertions(+)
 create mode 100644 Lab_07/Makefile
 create mode 100644 Lab_07/src/main.c
o71@071:~/Lvov_Artem/Lab_07$ git push
Username for 'https://github.com': sudo-071
Password for 'https://sudo-071@github.com':
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 1.06 KiB | 1.06 MiB/s, done.
Total 5 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/sudo-071/Lvov_Artem.git
   9a28cda..cd084a9  master -> master
o71@071:~/Lvov_Artem/Lab_07$
```

Рисунок 3 – Виконання команд git add, git commit, git push

4. Наведу графічну блок-схему роботи даної програми:

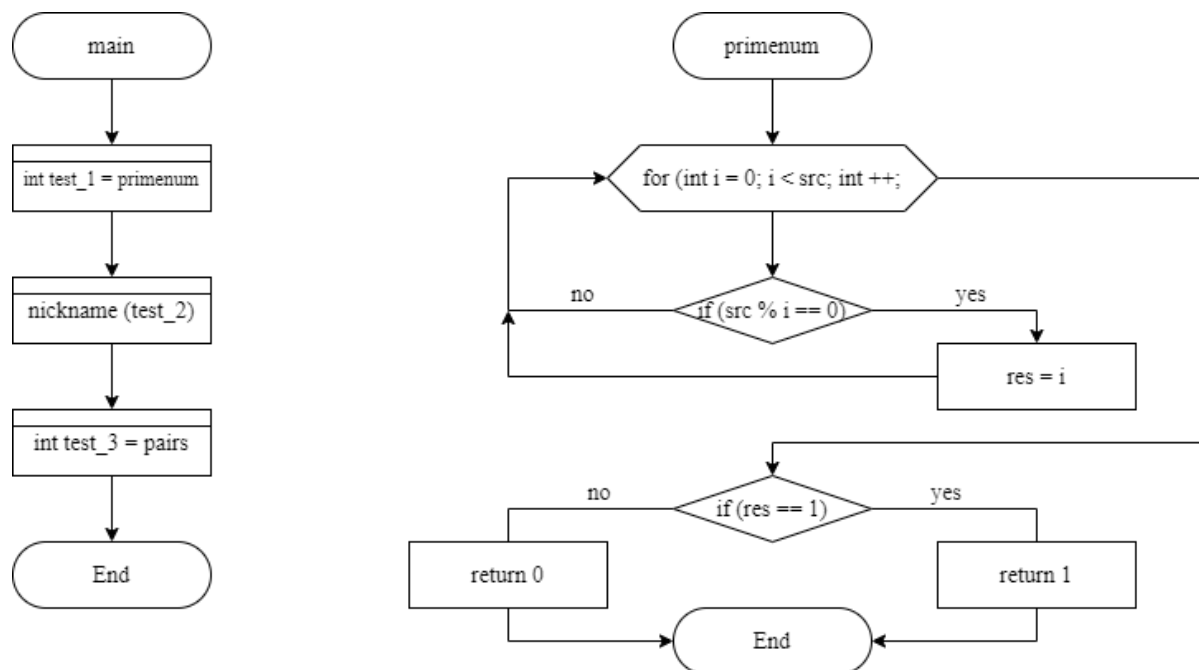


Рисунок 4.1 – блок-схема функцій main() та primenum()

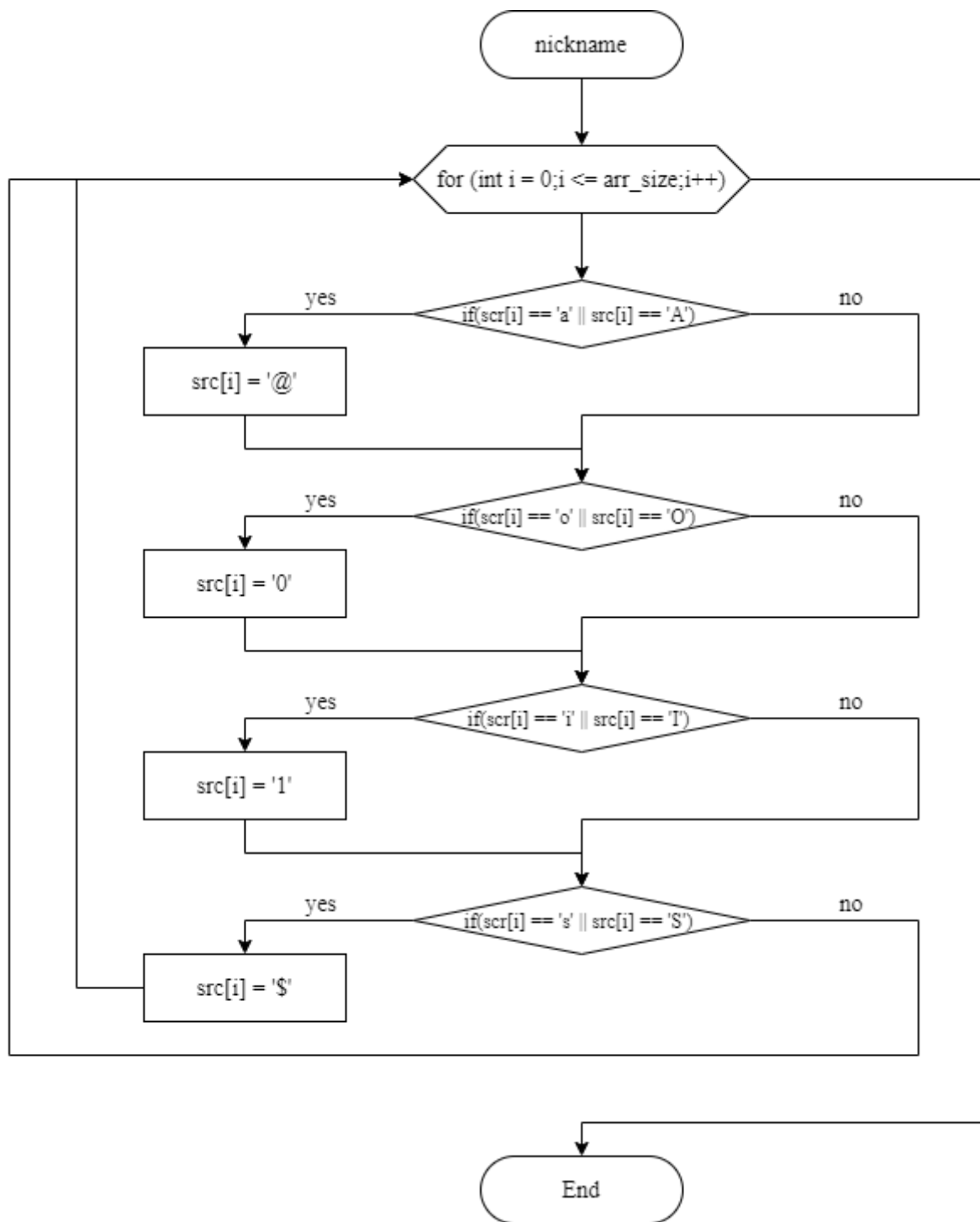


Рисунок 4.2 – блок-схема функції nickname()

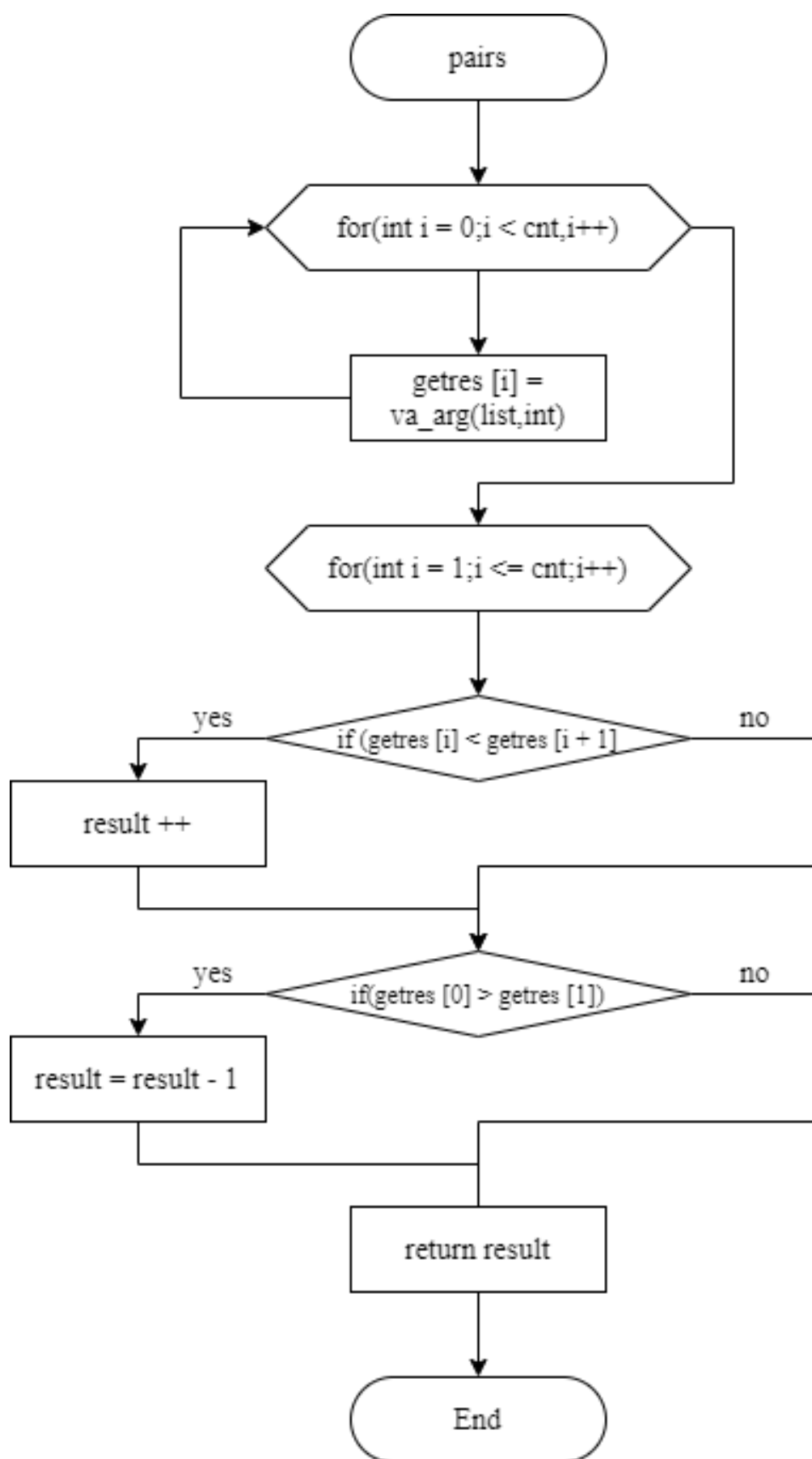


Рисунок 4.3 – блок-схема функції pairs()

Висновки:

Під час виконання цієї лабораторної роботи я навчився створювати функції, об'являти їх прототипи, викликати їх та навіть працювати з функціями зі змінним числом аргументів, бібліотекою `stdarg.h` та макросами з неї.