

Лабораторна робота №8

Вступ до документації коду

Мета: Переробити програми, розроблені під час виконання лабораторних робіт на теми

"Массивы" і "Цикли" таким чином, щоб код не мав повторень, використовуючи функції та написати додаткову програму

1 Вимоги

1.1 Розробник

Інформація про розробника:

- Соболєнко С. С.;

- КІТ-1216;

1.2 Загальне завдання

1) Переробити програми, розроблені під час виконання лабораторних робіт на теми

"Массивы" і "Цикли" таким чином, щоб код не мав повторень, використовуючи функції

2) Написати додаткову програму з використанням функцій

3) Продемонструвати роботу в режимі відлагодження

1.3 Задача

1. Переробити програму с lab05 використовуючи функції
2. Переробити програму с lab06 використовуючи функції
3. Визвати функції з використанням згенерованих параметрів
4. Написати програму що визначає, скільки серед заданої послідовності чисел таких пар, у котрих перше число менше наступного, використовуючи функцію

2. ОПИС ПРОГРАМИ

2.1 Функціональне призначення

1. Призначення перших двох програм не змінилось
2. Призначення нової програми: знаходження рішення алгоритму з масивом за допомогою арифметичних операцій, циклічної конструкції, умовної конструкції та функції. Результат зберігається у відповідній змінній. Демонстрація знайденого результату передбачає покрокове виконання програми в режимі відлагодження.

2.2 Опис логічної структури

- 1) Логіка перших двох програм не змінилась, лише програми була перероблені з використанням функцій та додано визов цих функцій зі згенерованими значеннями

2) Реалізую функцію, у якій циклом проходжу по всім членам послідовності, починаючи з другого, та завдяки умові у циклі з'ясовую чи є i -ий елемент меншим за $(i-1)$ -ий, якщо так, то збільшую кількість знайдених пар

Алгоритм програми наведено на рис.2.1.

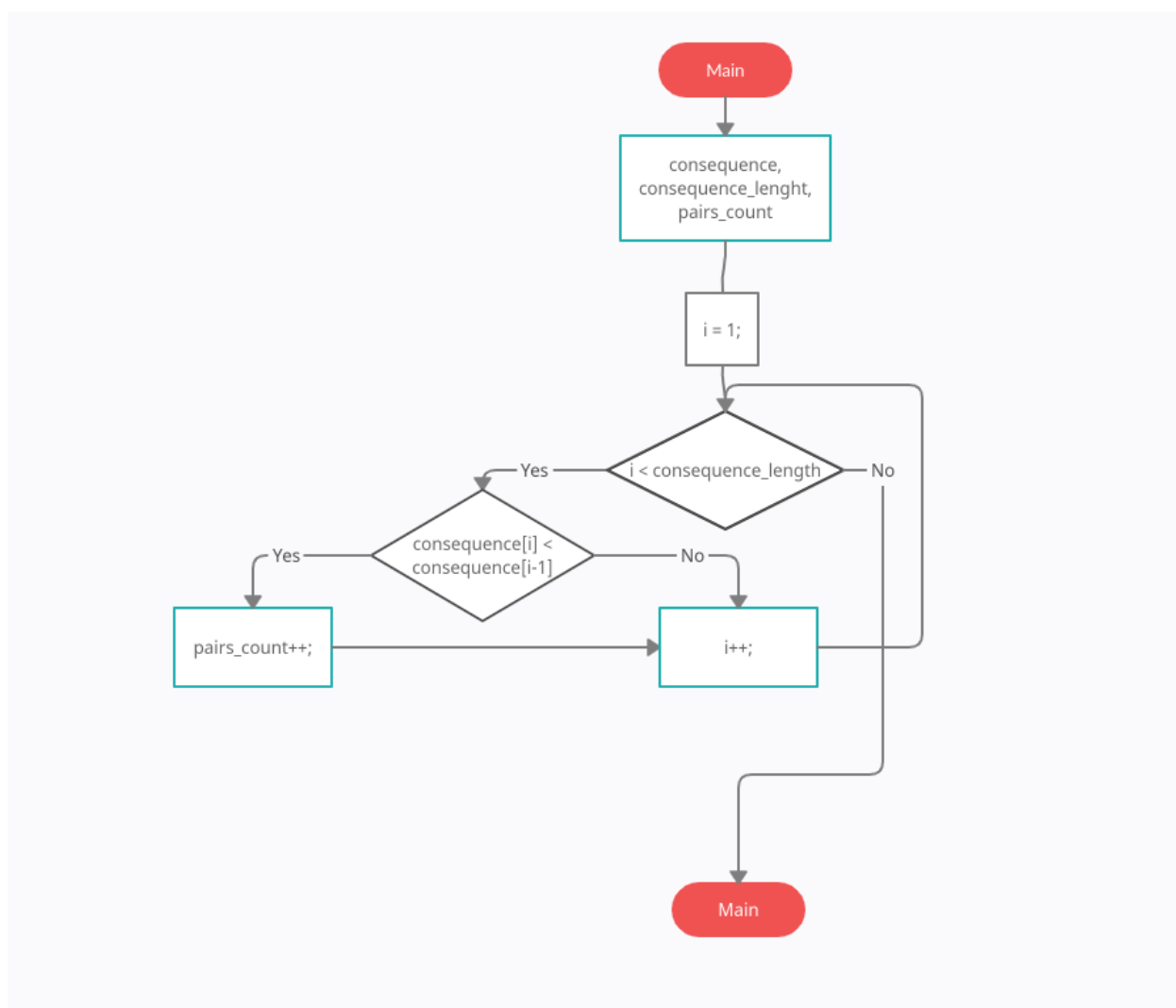


Рисунок 2.1 - Схема алгоритму програми

Структура проекту наведена на рис.2.2.

```
serga@serga-MS-7917:~/Documents/university/grade_1/semestr_1/programming/labs/lab07$ tree
.
├── dist
│   ├── addition_main.bin
│   ├── arrays_main.bin
│   └── cycles_main.bin
├── doc
│   └── assets
│       ├── addition_breakpoints.png
│       ├── addition_reult.png
│       ├── arrays_breakpoints.png
│       ├── arrays_result.png
│       ├── cycles_breakpoints.png
│       ├── cycles_result.png
│       ├── project_build.png
│       └── scheme.png
├── lab07.md
├── Makefile
├── README.md
└── src
    ├── addition
    │   └── main.c
    ├── arrays
    │   └── main.c
    └── cycles
        └── main.c

7 directories, 17 files
serga@serga-MS-7917:~/Documents/university/grade_1/semestr_1/programming/labs/lab07$ |
```

Рисунок 2.2 - Структура проекту

2.3 Важливі фрагменти програми

Об'явлення та встановлення початкових значень змінних

```
```c
```

```
 /* Задаю довжину послідовності */
```

```
 #define NUMBERS_COUNT 5
```

```
 /* Задаю послідовність чисел */
```

```
 int numbers [NUMBERS_COUNT] = {3, 2, 4, 3, 1};
```

```
 /* Результуюча кількість пар у яких перше число більше другого */
```

```
 int result = get_pairs_count(numbers, NUMBERS_COUNT);
```

```
```
```

Підрахунок кількості пар які задовольняють вказаній умові

```
```c
```

```
int get_pairs_count(int consequence[], int consequence_length) {
```

```
 /* Задаю лічильник кількості знайдених пар */
```

```
 int pairs_count = 0;
```

```
 /* Проходжусь по всім числам послідовності починаючи з другого
для визначення кількості пар */
```

```
 for(int i = 1; i < consequence_length; i++) {
```

```
 /* Якщо i-те число менше ніж (i-1)-ше число, то збільшуємо
кількість пар */
```

```
 if(consequence[i] < consequence[i-1]) {
```

```
 pairs_count++;
```

```
 }
```

```
 }
```

```
 return pairs_count;
```

```
}
```

```
```
```

3 ВАРІАНТИ ВИКОРИСТАННЯ

Для демонстрації результатів використовується покрокове виконання

програми.

0) Збірка програми (див. рис.3.0)

```
serga@serga-MS-7917:~/Documents/university/grade_1/semestr_1/programming/labs/lab07$ make all
make clean prep compile run
make[1]: Entering directory '/home/serga/Documents/university/grade_1/semestr_1/programming/labs/lab07'
rm -rf dist
mkdir dist
clang -g src/arrays/main.c -o ./dist/arrays_main.bin
clang -g src/cycles/main.c -o ./dist/cycles_main.bin
clang -g src/addition/main.c -o ./dist/addition_main.bin
./dist/arrays_main.bin
./dist/cycles_main.bin
./dist/addition_main.bin
make[1]: Leaving directory '/home/serga/Documents/university/grade_1/semestr_1/programming/labs/lab07'
serga@serga-MS-7917:~/Documents/university/grade_1/semestr_1/programming/labs/lab07$ |
```

Рисунок 3.0 - Виконання Makefile

1) Установка точек зупину першої програми(див. рис.3.1.1)

```
serga@serga-MS-7917:~/Documents/university/grade_1/semestr_1/programming/labs/lab07$ lldb ./dist/cycles_main.bin
(lldb) target create "./dist/cycles_main.bin"
Current executable set to '/home/serga/Documents/university/grade_1/semestr_1/programming/labs/lab07/dist/cycles_main.bin' (x86_64)
.
(lldb) list
21  int main() {
22
23      srand(time(NULL));
24
25      /* Задаю значения данного числа */
26      int number = rand() % 10;
27      /* Оголошую змінну, що буду означати результат*/
28      bool result = is_number_great(number);
29
30      return 0;
(lldb) b -l 29
Breakpoint 1: where = cycles_main.bin`main + 62 at main.c:30:5, address = 0x000000000040118e
```

Рисунок 3.1.1 - Точки зупину, що встановлені

2) Покрокове виконання першої програми (див. рис.3.1.2).

```
(lldb) run
Process 66242 launched: '/home/serga/Documents/university/grade_1/semestr_1/programming/labs/lab07/dist/cycles_main.bin' (x86_64)
Process 66242 stopped
* thread #1, name = 'cycles_main.bin', stop reason = breakpoint 1.1
  frame #0: 0x000000000040118e cycles_main.bin`main at main.c:30:5
   27      /* Оголошую змінну, що буду означати результат*/
   28      bool result = is_number_great(number);
   29
->  30      return 0;
   31  }
   32
   33  bool is_number_great(int number){
(lldb) fr v
(int) number = 6
(bool) result = true
(lldb) |
```

Рисунок 3.1.2 - Результат у змінній "result"

3) Установка точек зупину другої програми(див. рис.3.2.1)

```
serga@serga-MS-7917:~/Documents/university/grade_1/semestr_1/programming/labs/lab07$ lldb ./dist/arrays_main.bin
(lldb) target create "./dist/arrays_main.bin"
Current executable set to '/home/serga/Documents/university/grade_1/semestr_1/programming/labs/lab07/dist/arrays_main.bin' (x86_64)
.
(lldb) list
22  int main() {
23
24      srand(time(NULL));
25
26      /* Задаю довжину тексту*/
27      #define TEXT_LEN 7
28      /* Задаю даний текст */
29      char text[TEXT_LEN];
30
31      /* Випадково заповнюю даний текст */
(lldb) b -l 40
Breakpoint 1: where = arrays_main.bin`main + 151 at main.c:41:5, address = 0x00000000004011e7
```

Рисунок 3.2.1 - Точки зупину, що встановлені

4) Покрокове виконання другої програми (див. рис.3.2.2).

```
(lldb) run
Process 66387 launched: '/home/serga/Documents/university/grade_1/semestr_1/programming/labs/lab07/dist/arrays_main.bin' (x86_64)
Process 66387 stopped
* thread #1, name = 'arrays_main.bin', stop reason = breakpoint 1.1
  frame #0: 0x00000000004011e7 arrays_main.bin`main at main.c:41:5
   38      /* Результуюча кількість слів */
   39      int result = get_words_count(text, TEXT_LEN);
   40
-> 41      return 0;
   42  }
   43
   44  int get_words_count(char string[], int string_len){
(lldb) fr v
(char [7]) text = "S XM  "
(int) result = 2
(lldb) |
```

Рисунок 3.2.2 - Результат у змінній "result"

5) Установка точок зупину нової програми(див. рис.3.3.1)

```
serga@serga-MS-7917:~/Documents/university/grade_1/semestr_1/programming/labs/lab07$ lldb ./dist/addition_main.bin
(lldb) target create "./dist/addition_main.bin"
Current executable set to '/home/serga/Documents/university/grade_1/semestr_1/programming/labs/lab07/dist/addition_main.bin' (x86_64).
(lldb) list
20  int main() {
21
22      /* Задаю довжину послідовності */
23      #define NUMBERS_COUNT 5
24      /* Задаю послідовність чисел */
25      int numbers [NUMBERS_COUNT] = {3, 2, 4, 3, 1};
26      /* Результуюча кількість пар у яких перше число більше другого */
27      int result = get_pairs_count(numbers, NUMBERS_COUNT);
28
29      return 0;
(lldb) b -l 28
Breakpoint 1: where = addition_main.bin`main + 68 at main.c:29:5, address = 0x0000000000401154
```

Рисунок 3.3.1 - Точки зупину, що встановлені

6) Покрокове виконання нової програми (див. рис.3.3.2).

```
(lldb) run
Process 66461 launched: '/home/serga/Documents/university/grade_1/semestr_1/programming/labs/lab07/dist/addition_main.bin' (x86_64)
Process 66461 stopped
* thread #1, name = 'addition_main.b', stop reason = breakpoint 1.1
  frame #0: 0x0000000000401154 addition_main.bin`main at main.c:29:5
   26      /* Результуюча кількість пар у яких перше число більше другого */
   27      int result = get_pairs_count(numbers, NUMBERS_COUNT);
   28
-> 29      return 0;
   30  }
   31
   32  int get_pairs_count(int consequence[], int consequence_length) {
(lldb) fr v
(int [5]) numbers = ([0] = 3, [1] = 2, [2] = 4, [3] = 3, [4] = 1)
(int) result = 3
(lldb) |
```

Рисунок 3.3.2 - Результат у змінній "result"

ВИСНОВКИ

Перероблено код з перших двох програм з використанням функцій, також до коду додано вивід цих функцій зі псевдорандомно згенерованими параметрами. Написано додаткову програму з використанням функцій. Засоби відлагодження та покрокове виконання програми дозволяють продемонструвати коректність реалізованого алгоритму.