

Лабораторна робота №7. Функції

1 Вимоги

1.1 Розробник

- Кузнецов Микита Олександрович
- студент групи КІТ-320
- 19-dec-2020

1.2 Загальне завдання

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

1.3 Індивідуальне завдання

Заповнити масив із заданої кількості елементів простими числами, що не повторюються. Розмір вихідного масиву задати наперед відомим значенням, що може бути більшим ніж результуюча кількість отриманих елементів.

Опис програми

1.4 Функціональне призначення

Програма призначена для заповнення масива із заданої кількості елементів простими числами, що декларовано в файлі `main.c`.

Результат заповнення зберігається у масиві *arrResult*

Демонстрація отриманих результатів передбачає покрокове виконання програми в режимі налагодження.

1.5 Опис логічної структури

За допомогою генератору псевдовипадкових чисел `rand` генеруємо значення для *ARRSIZE*, який вказує на розмір результуючого масива, через змінну *limit* вказуємо діапазон пошуку простих чисел.

Для отримання результату використовується функція `arrWithPrimeNum`, що знаходиться в `main.c`

Структура проекту

```
. |— README.md
  |— src
    |— main.c
```

1.6 Важливі фрагменти програми

Генерація псевдовипадкового числа.

```
srand(time(0));
```

```
int arrSize = rand() % 45 + 15; // Размер массива в диапазоне от 15 до 45 символовчисел
```

Пошук та запис простих чисел в масив

```
for (int i = 2; i < limit; i++) {
    for (int j = 2; j < i; j++) {
        if (i % j == 0) {
            check = 1; // Если check = 1 - число не простое
            break;
        } else {
            check = 0; // Если check = 0 - число простое
        }
    }

    if (check == 0) { // Число простое - записываем его в массив, если число не простое - пропускаем
        arrResult[k] = i;
        k++;
    }
}
```

2 Варіанти використання

Для демонстрації результатів використовується покрокове виконання програми в інтегрованому середовищі *Nemiver*. Нижче наводиться послідовність дій запуску програми у режимі відлагодження.

Крок 1 (див рис. 1). Знаходячись в основній процедурі, досліджуємо стан масива

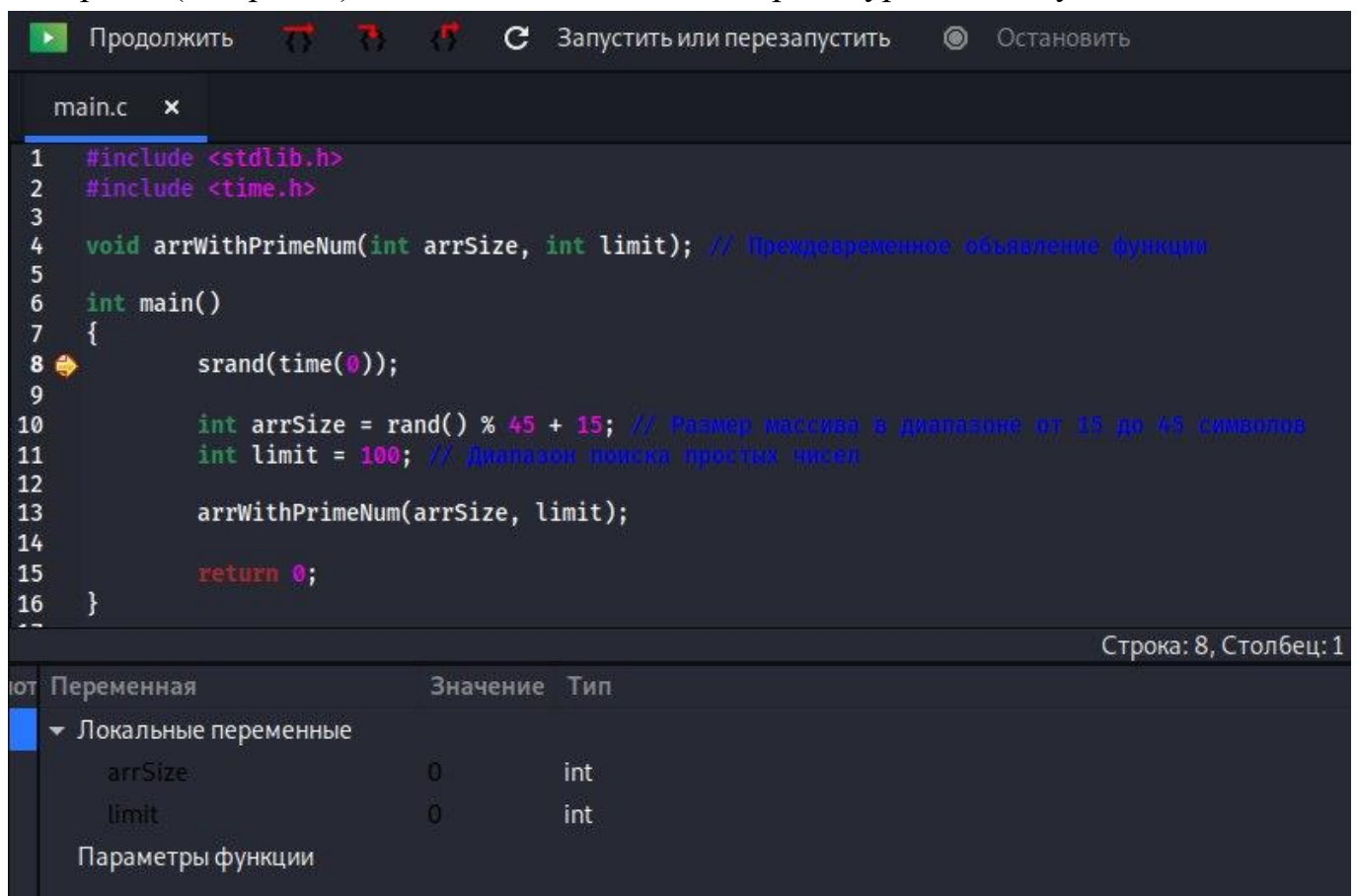


Рисунок 1 – вікно відлагодження в основній процедурі

Крок 2. Дослідження стану масива наприкінці виконання основної функції. Результат зображено на рис. 2, результат пошуку та запису простих чисел можна побачити в масиві *arrResult*

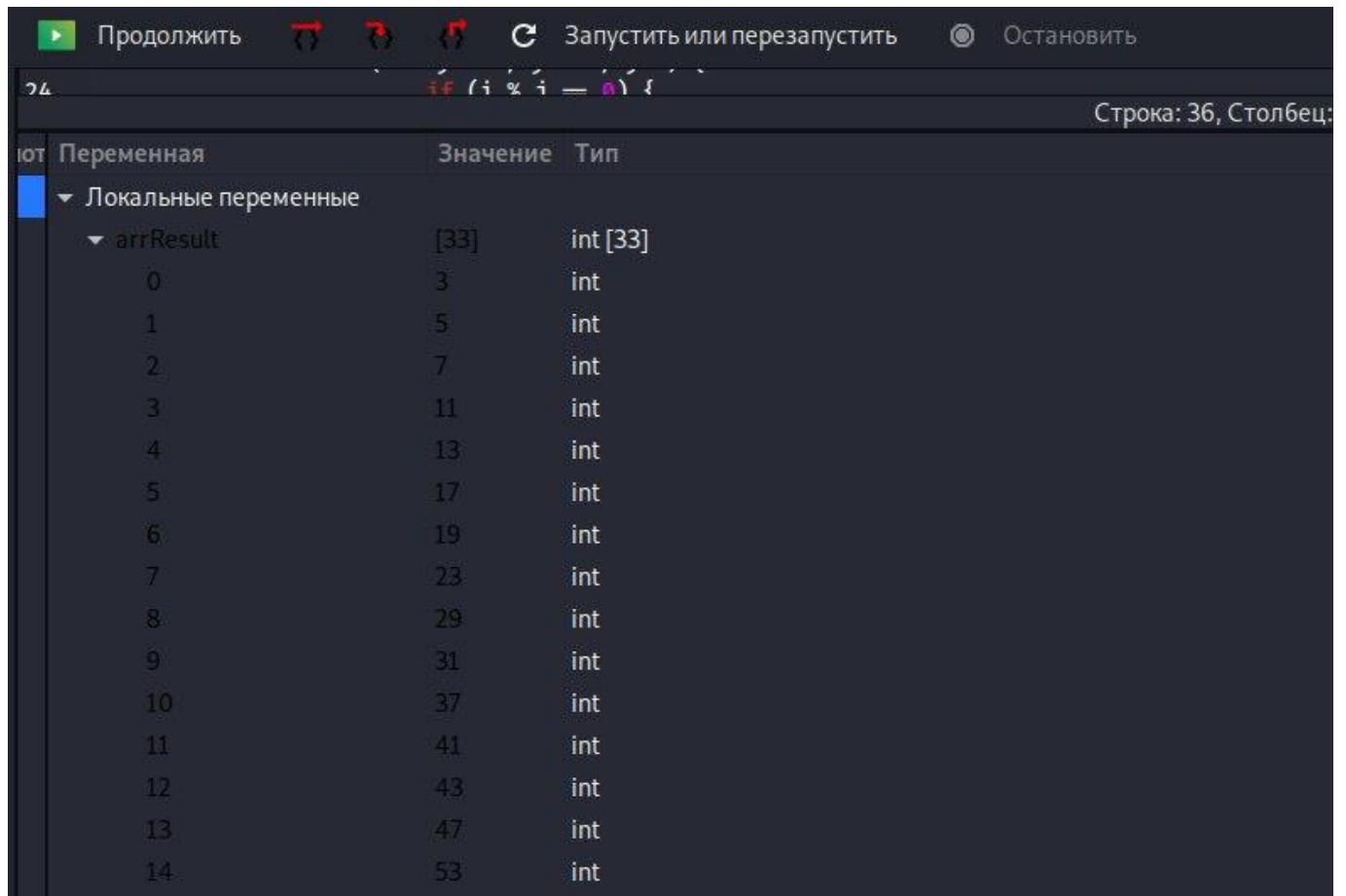


Рисунок 2 – вікно відлагодження з результатом

3 Висновки

При виконанні даної лабораторної роботи було набуто практичного досвіду роботи з функціями на генератором псевдовипадкового числа rand.