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

Визначити, чи є задане число досконалим (якщо воно дорівнює сумі своїх дільників). Наприклад, 6 – досконале число, бо 6 = 1 + 2 + 3.

Опис програми

1.4 Функціональне призначення

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

Результат обчислення зберігається у змінній *perfectnumResultFor, perfectnumResultWhile, perfectnumResultDoWhile*

Демонстрація отриманих результатів передбачає покрокове виконання програми в режимі налагодження.

1.5 Опис логічної структури

За допомогою генератору псевдовипадкових чисел rand генеруємо значення для *num*, Для отримання результату використовується функція perfectnum_for, perfectnum_while, perfectnum_dowhile що знаходиться в *main.c*

Структура проекту

```
. ├── README.md
  └── src
    └── main.c
```

1.6 Важливі фрагменти програми

Генерація псевдовипадкового числа.

```
srand(time(0));

int num = rand() % 99 + 1; // Размер массива в диапазоне от 15 до 45 символовчисел
```

Обчислення через функції perfectnum_for:

```
int sumFor = 0;
int resultFor;
int tempFor;
for (int i = 1; i < num; i++) {
	tempFor = num % i;
	if (tempFor == 0) {
		sumFor += i;
	}
}
if (sumFor == num) {
	resultFor = 0; // 0 - Число совершенное
} else if (sumFor != num) {
	resultFor = 1; // 1 - Число не совершенное
}
return resultFor;
```

Обчислення через функції perfectnum_while:

```
int sumWhile = 0;
int resultWhile;
int tempWhile;
int j = 1;
while (j < num) {
	tempWhile = num % j;
	if (tempWhile == 0) {
		sumWhile += j;
	}
	j++;
}
if (sumWhile == num) {
	resultWhile = 0; // 0 - Число совершенное
} else if (sumWhile != num) {
	resultWhile = 1; // 1 - Число не совершенное
}
return resultWhile;
```

Обчислення через функції perfectnum_dowhile:

```
int sumDoWhile = 0;
int resultDoWhile;
int tempDoWhile;
int k = 1;
do {
	tempDoWhile = num % k;
	if (tempDoWhile == 0) {
		sumDoWhile += k;
	}
	k++;
} while (k < num);
if (sumDoWhile == num) {
	resultDoWhile = 0; // 0 - Число совершенное
}else if (sumDoWhile != num) {
	resultDoWhile = 1; // 1 - Число не совершенное
}
return resultDoWhile;
```

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 основної процедурі, досліджуємо стан масива

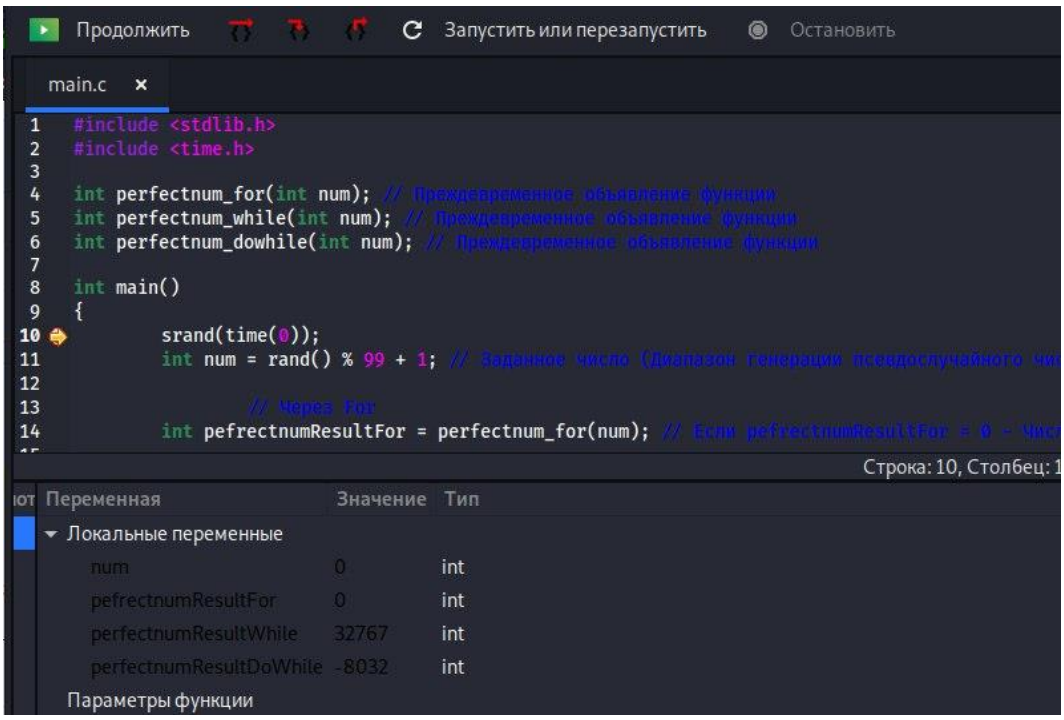

Рисунок 1 – вікно відлагодження в основної процедурі

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*

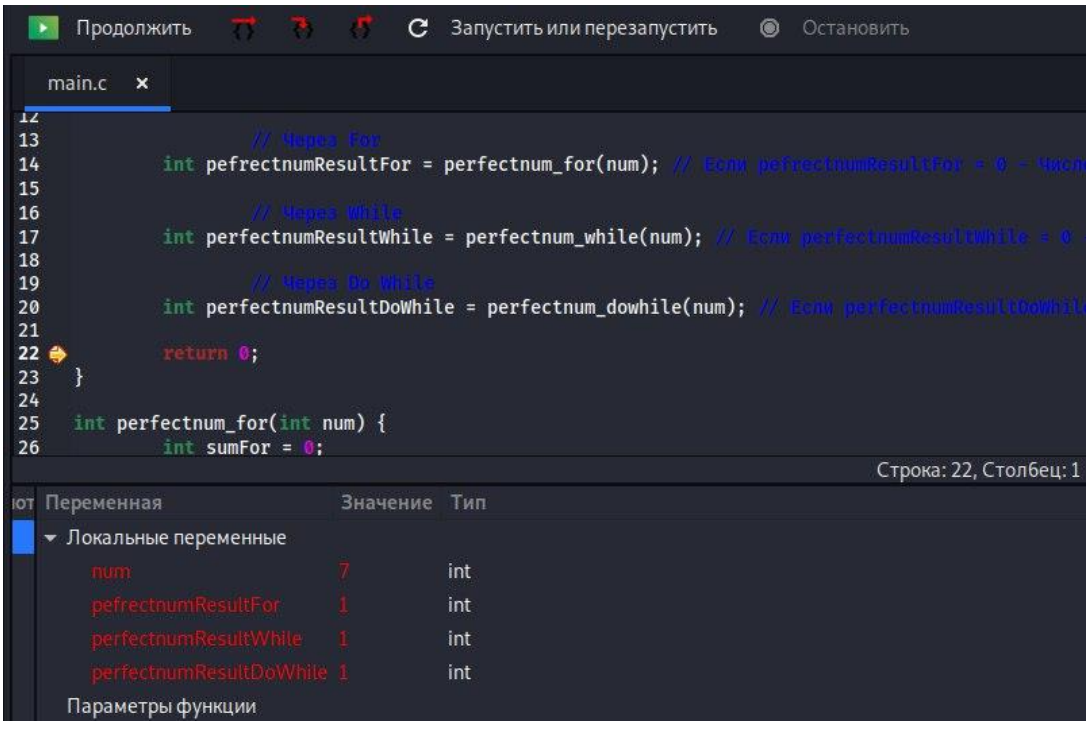

Рисунок 2 – вікно відлагодження з результатом

2 Висновки

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