

Лабораторна робота №4. Розробка програм, що розгалужуються

1 Вимоги

1.1 Розробник

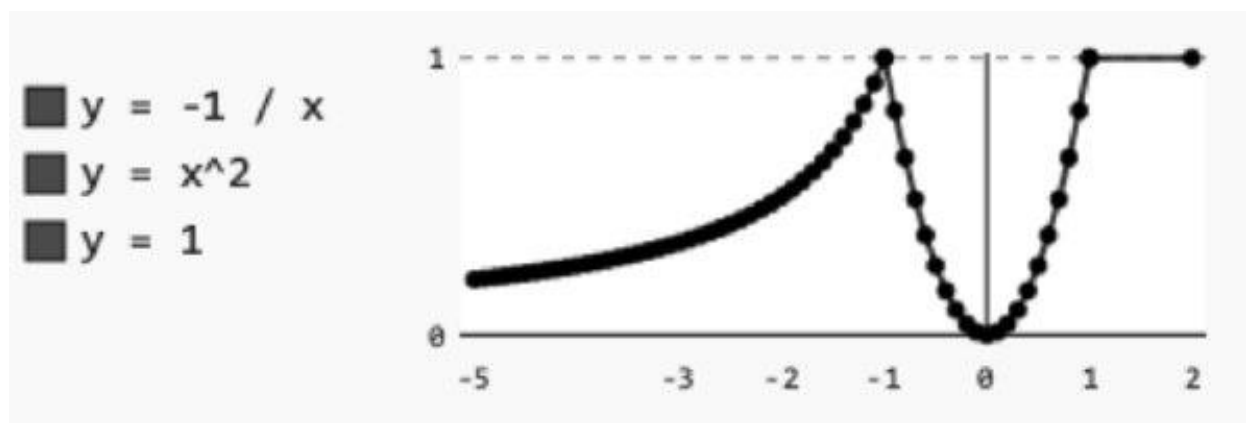
- Кузнецов Микита Олександрович
- студент групи КІТ-320
- 15-nov-2020

1.2 Загальне завдання

Розробити програму, яка розгалужується (прикладна галузь вказана в індивідуальному завданні)

1.3 Індивідуальне завдання

Дано дійсне число x . Для функції $y(x)$, графік якої подано на рycinку нижче, обчислити значення y .



Опис програми

1.4 Функціональне призначення

Програма призначена для визначення значення y , якщо дано дійсне число x . Для функції $y(x)$, графік якої подано на рисунку 1, що декларовано в файлі *ScheduleYx.c*.

Результат зберігається у змінній y .

Демонстрація отриманих результатів передбачає покрокове виконання програми в режимі налагодження.

1.5 Опис логічної структури

За допомогою `#define` задаємо початкові дані для X .

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

Структура проекту

```
.
├── dist
│   └── ScheduleYx.bin
├── Doxyfile
├── Makefile
├── README.md
└── src
    └── ScheduleYx.c
```

1.6 Важливі фрагменти програми

Початкові дані. Define

```
#define X -1.5 // Число X
```

Обчислення відстані

```
float y = 0;

if (X <= -1) {
    y = -1 / X;
}
else if (X > -1 && X <= 1) {
    y = X * X;
}
else if (X > 1) {
    y = 1;
}
```

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 змінних

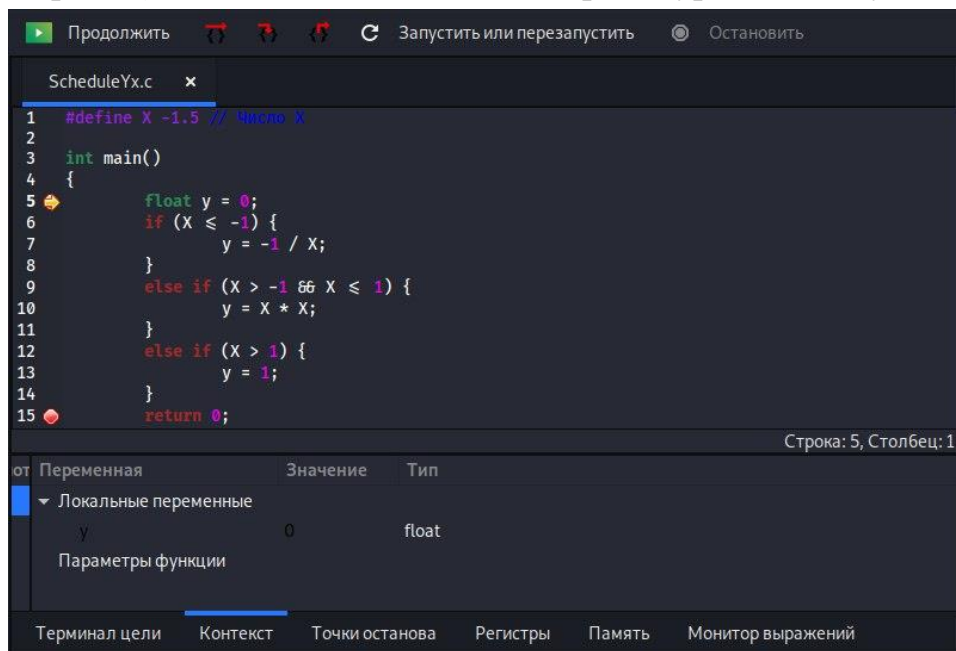


Рисунок 1 – вікно відлагодження в основній процедурі

Крок 2. Дослідження стану змінних наприкінці виконання основної функції. Результат зображено на рис. 2, результат обчислення відстані можна побачити у змінній *y*.

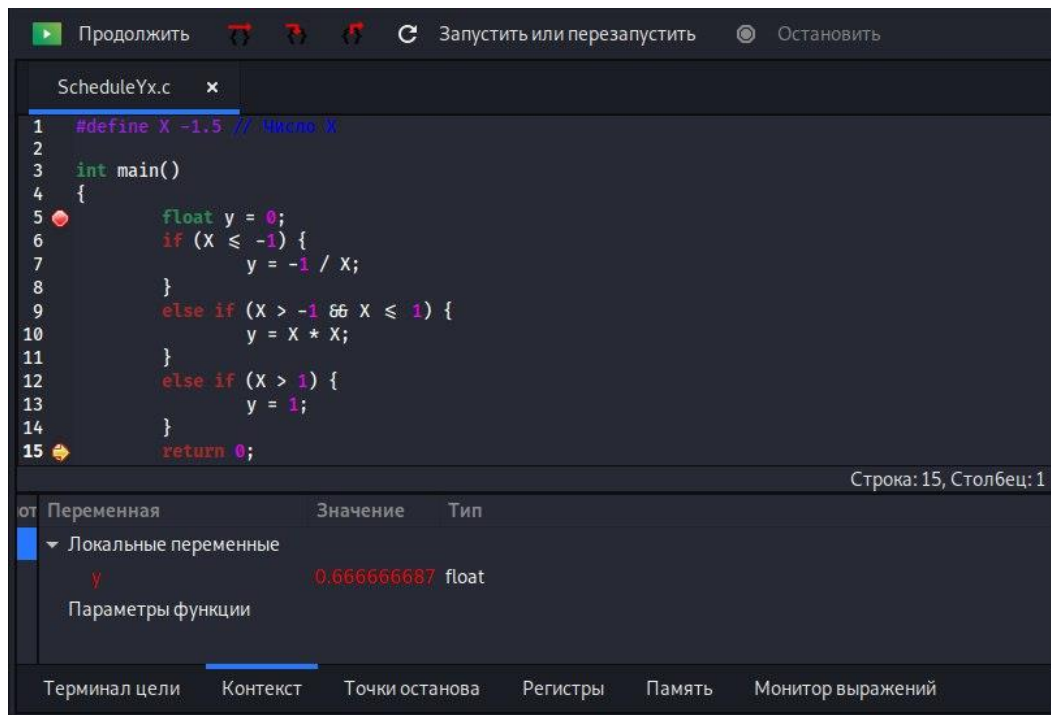


Рисунок 2 – вікно відлагодження з результатом

3 Висновки

При виконанні даної лабораторної роботи було набуто практичного досвіду роботи з `if`.