

Отчет: Лабораторная работа №2

УРОК ИНФОРМАТИКИ В 9А

САДРЕТДИНОВ КИРИЛЛ РУСЛАНОВИЧ СКБ241_2

1. В ходе лабораторной работы я написал 3 функции(двойной факториал, и две линейных функции вычисления), изучил необходимые для реализации библиотеки (cmath), опубликовал код в репозиторий на GitHub.2.

2. Сведения о системе

Операционная система: macOS Sequoia 15.0

Компилятор: Apple clang version 15.0.0 (clang-1500.3.9.4). Является частью Command line tools, из-за чего имеет нативную поддержку компиляции под ARM. Схож с Clang

IDE: Visual Studio Code 1.93. С самого начала всегда пользовался только VSC, очень удобен, имеет большое количество расширений (pylance, docker, thunderclient, tabnine), поддерживает все современные языки программирования, очень удобная отладка.

Git version 2.39.3 (Apple Git-146)

GitHub Desktop 3.4.5

3. Процедура

Для начала я разделил задачу на 3 логические части: функция двойного факториала, функция $t(x)$ и основная функция, в которой содержится ввод и конечная формула. Для вычисления двойного факториала я написал условную конструкцию для трех случаев ($n=0$, n нечетное, n четное), подсчет факториала с помощью цикла for. Для функции $t(x)$ написал два цикла for (для каждого из знаков сигма), а также использовал функцию `pow()` из библиотеки `cmath` для операции возведения в степень. В основной функции объявил две переменные: для ввода входного значения и для подсчета итогового, также использовал `pow()`. Во всей программе используются типы данных, занимающее максимально возможный объем памяти, чтобы была возможность работать с большими числами.

4. Код (GitHub)

github.com/hosternus/cpp-lab-2

5. Заключение

Во время работы я познакомился с новой функцией (`pow()`), столкнулся с проблемой, когда мой алгоритм выводил неверный результат, осознал, для чего нужны большие типы данных (`long long`, `double`), научился работать с циклами for.

6. Приложение

```
6  ~ int64_t double_factorial(int64_t y) {
7      if (y <= 0) {
8          return 1;
9      } else if ((y % 2) == 0) {
10         int64_t res = 1;
11         for (long i = 2; i <= y; i += 2) {
12             res *= i;
13         }
14         return res;
15     } else {
16         int64_t res = 1;
17         for (long i = 1; i <= y; i += 2) {
18             res *= i;
19         }
20         return res;
21     }
22 }
```

```
26 ~ long double t(long double x) {
27     long double sum1, sum2 = 0;
28
29     for (int i = 0; i <= 10; i++) {
30         sum1 += (pow(x, ((2 * i) + 1))) / (double_factorial((2 * i) + 1));
31     }
32
33     for (int i = 0; i <= 10; i++) {
34         sum2 += (pow(x, (2 * i))) / (double_factorial(2 * i));
35     }
36
37     return (sum1 / sum2);
38 }
39 }
```

```
42 ~ int main(void) {
43
44     long double y, res;
45
46     cin >> y;
47
48     res = ((7 * t(0.25)) + (2 * t(1 + y))) / (6 - (t(pow(y, 2) - 1)));
49
50
51     cout << res << endl;
52
53     return 0;
54 }
```