

Отчет по лабораторной работе № 9 по курсу Фундаментальная информатика

Студент группы: М8О-101Б-22, Кабанов Антон Алексеевич, № по списку: 7, контакты: anton1258kab@gmail.com

Работа выполнена: "8" ноября 2022 г.

Преподаватель: каф. 806 Крылов Сергей Сергеевич

Входной контроль знаний с оценкой:

Отчет сдан "17" ноября 2022 г., итоговая оценка

Подпись преподавателя:

1. **Тема:** Составление простейшей программы на языке C
2. **Цель работы:** Составление и отладка простейшей программы на языке C итеративного характера с целочисленными рекуррентными соотношениями, задающими некоторое регулярное движение точки в целочисленной системе координат (i, j) с дискретным временем k и динамическим параметром движения l
3. **Задание (вариант № 7):** Написание программы на языке C согласно условию (цели) лабораторной работы по некоторому закону, выводящую результат попадания или промаха:
Квадрат с длиной стороны 10, стороны квадрата параллельны осям координат, центр квадрата в точке $(10, -10)$:
 $i_0 = 11, j_0 = 13, l_0 = 10; i_{k+1} = |k - 15| - \min(\frac{i_k}{3}, (j_k + l_k) \bmod 10) - 20; j_{k+1} = -(j_k + k)/5 + |i_k l_k \bmod 8|; l_{k+1} = \max((i_k + j_k) \bmod 15, (l_k + k) \bmod 14).$
4. **Оборудование:**
Оборудование ПЭВМ студента, если использовалось:
Процессор AMD Ryzen 5500U (6-ядерный, @2.1 ГГц) с ОП 15345 Мб, ТТН 479.9 Гб. Монитор встроенный, IPS, 2160x1440, @60 Гц.
5. **Программное обеспечение (лабораторное):**
Программное обеспечение ЭВМ студента, если использовалось:
Операционная система семейства GNU/Linux, наименование Manjaro Linux версия 5.15.76-1-MANJARO, интерпретатор команд bash версия 5.1.16.
Система программирования: C
Редактор текстов: emacs, vim (neovim)
Утилиты операционной системы: pwd, who, ls, cd, mv, cp, rm, rmdir, mkdir, cat, whoami, man
Прикладные системы и программы: touch, echo, pacman, chmod, date, lsblk, gnuplot, emacs, nvim
Местонахождение и имена файлов программ и данных на домашнем компьютере: /home/void/Документы/FI-labs
6. **Идея, метод, алгоритм** решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)
7. **Сценарий выполнения работы** [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты либо соображения по тестированию]
Пункты 1-7 отчета составляются строго до начала лабораторной работы.

Допущен к выполнению работы. Подпись преподавателя:

8. **Распечатка протокола** (подклеить листинг окончательного варианта программы с тестовыми примерами, подписанный преподавателем):

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
```

```
4 int max(int x, int y) {
5     if (x <= y) {
6         return y;
7     } else {
8         return x;
9     }
10 }
11
12 int min(int x, int y) {
13     if (x <= y) {
14         return x;
15     } else {
16         return y;
17     }
18 }
19
20 int modulo(int x, int y) {
21     int k = 0;
22     if (x < 0) {
23         if (abs(x) % abs(y) == 0) {
24             k = 0;
25         } else {
26             k = 1;
27         }
28     }
29     return x - (x / y - k) * y;
30 }
31
32 int main() {
33     const int i0 = 11;
34     const int j0 = 13;
35     const int l0 = 10;
36     //printf("%d\n", modulo(23, 10));
37     int i_old = i0, j_old = j0, l_old = l0;
38     int i_next, j_next, l_next;
39     int k = 0;
40     int flag = 0;
41     while (1) {
42         i_next = abs(k - 15) - min(i_old / 3, modulo((j_old + l_old), 10)) - 20;
43         j_next = -(j_old + k) / 5 + abs(modulo(i_old * l_old, 8));
44         l_next = max(modulo(i_old + j_old, 15), modulo(l_old + k, 14));
45         i_old = i_next;
46         j_old = j_next;
47         l_old = l_next;
48         k++;
49         if ((i_old > 5 && i_old < 15) && (j_old > -15 && j_old < -5)) {
50             printf("it's in! x: %d, y: %d, step: %d\n", i_old, j_old, k);
51             break;
52         }
53         if (k == 50) {
54             flag = 1;
55             break;
56         }
57     }
58     if (flag == 1) {
59         printf("miss!\n");
60     }
61     return 0;
62 }
```

9. **Дневник отладки** должен содержать дату и время сеансов отладки и основные события (ошибки в сценарии и программе, нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы:

№	лаб/дом	Дата	Время	Событие	Действие по исправлению	Примечание

10. **Замечания автора** по существу работы: Можно было написать цикл **for (int i = 1; i <= 50; i++)** вместо **while (1)** с использованием переменной **flag**.

11. **Выводы:** Я научился писать простейшие программы на языке Си.

Недочёты при выполнении задания могут быть устранены следующим образом: -

Подпись студента: