

# Отчет по лабораторной работе № 2

## по курсу "Фундаментальная информатика"

Выполнил студент группы М8О-114БВ-24: Дробышев Егор Павлович, № по списку 29

Контакты e-mail: [tru.899@yandex.ru](mailto:tru.899@yandex.ru)

Работа выполнена: «15» ноября 2024 г.

Преподаватель: доцент каф. 806 Никулин Сергей

Петрович

Отчет сдан « » 2024г., итоговая оценка \_\_\_\_\_

Подпись преподавателя \_\_\_\_\_

**1. Тема:** Программирование на Си

**2. Цель работы:** Составление и отладка простейшей программы на языке С

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:** Эллипс с центром в точке  $(20, 0)$  и проходящий через точки  $(10, 0)$ ,  $(30, 0)$ ,  $(20, 5)$ ,  $(20, -5)$

$$\begin{aligned} 29. \quad & i_0 = -7, j_0 = -19, l_0 = 4 \\ & i_{k+1} = \max(i_k j_k, i_k l_k, j_k l_k) \bmod 30 + k, \\ & j_{k+1} = |j_k - l_k| \operatorname{sign} i_k - |i_k - l_k| \operatorname{sign} j_k, \\ & l_{k+1} = \min(i_k, \max(j_k, \min(l_k, \max(i_k - l_k, j_k - l_k)))) \end{aligned}$$

(Вариант – 29)

**4. Оборудование:** оборудование ПЭВМ студента, если использовалось:

Процессор \_Intel Core i5\_ с ОП 8 Гб НМД \_\_\_\_256\_\_\_\_Гб. Монитор

1920x1080~60Hz. Другие устройства не использовались

**5. Программное обеспечение:** программное обеспечение ЭВМ студента:

Операционная система семейства \_Linux\_, наименование \_Ubuntu\_ версия \_24.04 \_\_\_\_

интерпретатор команд \_GNU bash\_ версия \_5.2.21(1)\_.

Редактор текстов emacs версия 29.3

Утилиты операционной системы: gcc, gdb

Местонахождение и имена файлов программ и данных на домашнем компьютере:

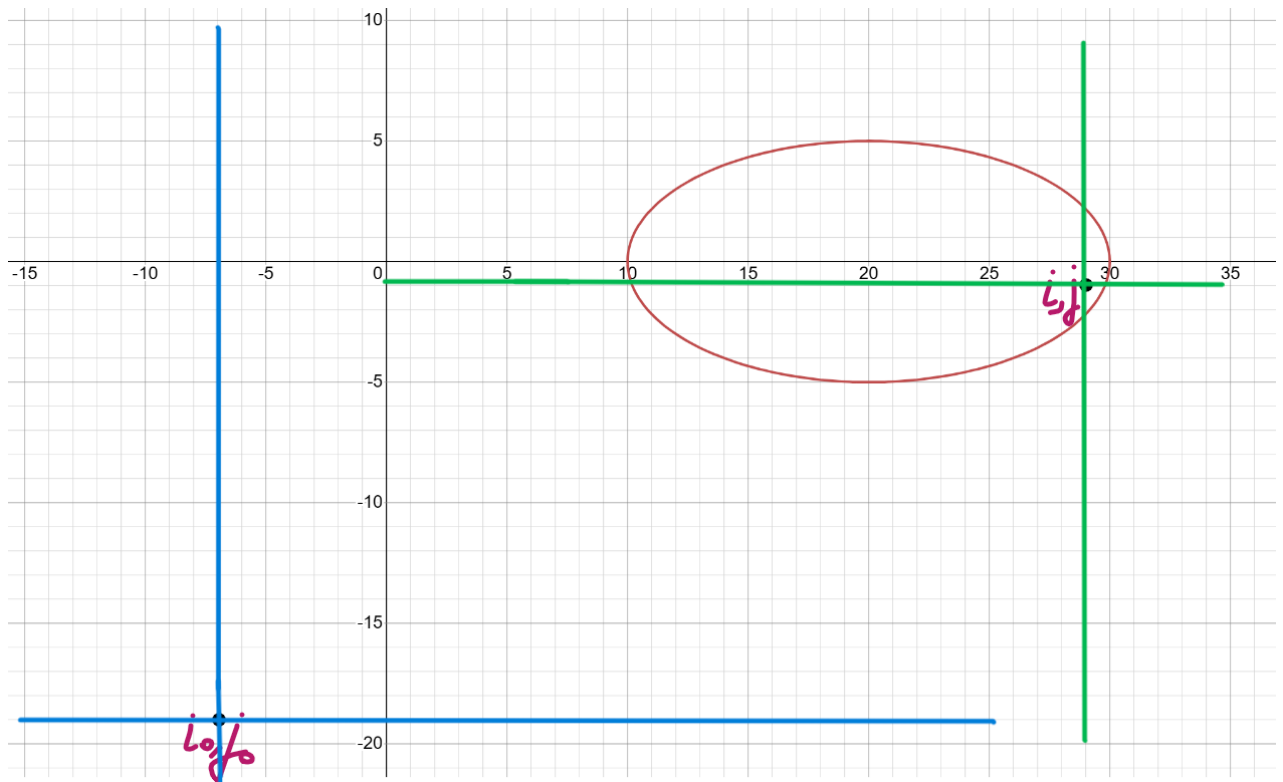
/home/tru\_\_

**6. Идея, метод, алгоритм решения задачи** (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)

Задаются начальные значения для переменных (координаты, параметры движения, шаги и т.д.). В каждом шаге вычисляются новые значения для координат  $i$  и  $j$ , а также для динамического параметра  $l$ . Для обновления координат используются функции для нахождения максимума, минимума, модуля, знака чисел и остатка от деления. Условие проверки попадания в эллиптическую область с центром в точке  $(x_0, y_0)$  и размерами, определяемыми параметрами  $a$  и  $b$ :

$$\text{уравнение эллипса: } (x_1 - x_0)^2/a^2 + (y_1 - y_0)^2/b^2 = 1$$

Если в какой-то момент точка попадает в эту область (уравнение эллипса  $\leq 1$ ), то алгоритм завершает выполнение и выводит информацию о времени завершения и конечных координатах. После выполнения или остановки выводится информация о достижении области и соответствующие параметры.



**7. Сценарий выполнения работы** [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты либо соображения по тестированию].

```
#include <stdio.h>

int max(int a, int b) {
    if (a >= b) {
        return a;
    } else {
        return b;
    }
}

int min(int a, int b) {
    if (a < b) {
        return a;
    } else {
        return b;
    }
}

int module(int a) {
    if (a >= 0) {
        return a;
    } else {
```

```

        return -a;
    }
}

int sign(int a) {
    if (a > 0) {
        return 1;
    } else if (a == 0) {
        return 0;
    } else {
        return -1;
    }
}

int mod(int a, int b) {
    return (a%b+b) % b;
}

int main() {
    const int i0 = -7, j0 = -19, l0 = 4, x0 = 20, y0 = 0, a = 10, b = 5;
    int i = i0, j = j0, l = l0, i1, j1, l1, k = 1, f = 0;
    for (k; k<=50; k++) {
        i1 = (mod((max(max(i * j, i * l), j * l)), 30)) + (k - 1);
        j1 = (module(j - l)) * sign(i) - (module(i - l) * sign(j));
        l1 = (min(i, max(j, min(l, max(i - l, j - l)))));
        i = i1;
        j = j1;
        l = l1;
        printf("%s%d %s%d %s%d %s%d\n", "k=", k, "l=", l, "i=", i, "j=",
j);
        if (((float)((i - x0) * (i - x0)) / (float)(a * a)) + ((float)((j
- y0) * (j - y0)) / (float)(b * b)) <= 1) {
            f = 1;
            break;
        }
    }
    if (f > 0) {
        printf("%s\n", "Попадание в заданную в область не более чем за 50
шагов");
        printf("%s%d\n", "Время окончания движения: ", k);
        printf("%s%d\n", "Значение динамического параметра движения: ", l);
        printf("%s%d%s%d%s\n", "Координаты конечной точки:
(", i, ",", j, ")");
    } else {
        printf("%s\n", "НЕ попадание в заданную в область не более чем за
50 шагов");
        printf("%s%d\n", "Время окончания движения: ", k);
        printf("%s%d\n", "Значение динамического параметра движения: ", l);
        printf("%s%d%s%d%s\n", "Координаты конечной точки: (", i, ",", j, ")");
    }
}
}

```

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

PS C:\c proj\vsc> .\laba9

k=1 l=-11 i=13 j=-12

k=2 l=-11 i=13 j=25

k=3 l=13 i=27 j=12

k=4 l=13 i=24 j=-13

k=5 l=11 i=16 j=37

k=6 l=16 i=27 j=21

k=7 l=21 i=33 j=-6

k=8 l=12 i=10 j=39

k=9 l=10 i=26 j=25

k=10 l=25 i=29 j=-1

Попадание в заданную в область не более чем за 50 шагов

Время окончания движения: 10

Значение динамического параметра движения: 25

Координаты конечной точки: (29,-1)

**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. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание
---	---------------------	------	-------	---------	-------------------------	------------


#### 10. Замечания автора по существу работы

---



---



---

#### 11. Выводы:

В ходе выполнения лабораторной работы я выучил и освоил работу в Си. Недочёты при выполнении задания могут быть устранены следующим образом:

---



---



---

Подпись студента\_\_\_\_\_