

# Отчет по лабораторной работе № 9 по курсу “Фундаментальная информатика”

Студент группы М80-109Б-22 Филиппов Александр Михайлович, № по списку 21

e-mail [a.filipov04@yandex.ru](mailto:a.filipov04@yandex.ru)

telegram @otaku0101

Работа выполнена: «12» ноябрь 2022г.

Преподаватель: каф. 806 Сысоев Максим Алексеевич

Отчет сдан «    » \_\_\_\_\_ 20\_\_ г., итоговая оценка \_\_\_\_\_

Подпись преподавателя \_\_\_\_\_

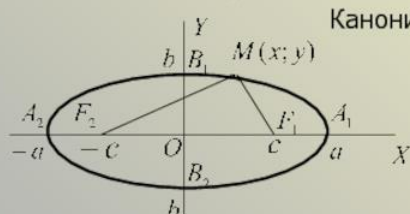
1. **Тема:** Язык программирования Си
2. **Цель работы:** Составление итерационного алгоритма, определяющего принадлежность точки, лежащей на координатной плоскости, заданной области
3. **Задание (вариант № 27):** Область задана эллипсом, проходящим через точки (10, 0), (30, 0), (20, 5) и (20, -5)
4. **Оборудование (студента):**  
Процессор *Intel Core i5-8265U @ 8x 3.9GH* с ОП 7851 Мб, НМД 1024 Гб. Монитор 1920x1080
5. **Программное обеспечение (студента):**  
Операционная система семейства: *linux*, наименование: *ubuntu*, версия *18.10 cosmic*  
интерпретатор команд: *bash* версия *4.4.19*.  
Система программирования -- версия --, редактор текстов *emacs* версия *25.2.2*  
Утилиты операционной системы --  
Прикладные системы и программы --  
Местонахождение и имена файлов программ и данных на домашнем компьютере --

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

Есть начальные координаты точки (i, j) и дополнительный параметр (l), которые с каждым из 50 шагов (k) меняются в соответствии с заданными уравнениями. На каждой новой итерации мы вычисляем новые i, j и l и проверяем, попала ли точка в область. Область представляет собой эллипс, проверить принадлежность точки к нему можно подставив ее координаты в каноническое уравнение эллипса. Если полученное значение равно единице, точка лежит на контуре фигуры, если меньше – внутри нее, если больше – за ее пределами.

## 2. Эллипс

**Определение.** Эллипсом называется множество точек плоскости, сумма расстояний которых до двух данных точек, называемых фокусами, есть величина постоянная, равная длине большой оси  $2a$ .



Каноническое уравнение эллипса

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1, \text{ причем}$$

$$a^2 = b^2 + c^2$$

$A_1(a; 0)$   $A_2(-a; 0)$  вершины эллипса  
 $B_1(0; b)$   $B_2(0; -b)$

$F_1(c; 0)$   $F_2(-c; 0)$

фокусы эллипса

$A_1A_2 = 2a$  большая ось эллипса  
 $B_1B_2 = 2b$  малая ось эллипса

$F_1F_2 = 2c$  фокусное расстояние

!

В уравнение эллипса входят квадраты переменных, причем знаки при квадратах одинаковые, а коэффициенты при квадратах разные.

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

Основная сложность заключается в запрете на использование встроенной математической библиотеки `math.h`, поэтому большую часть времени занимает реализация нужных функций, использующихся в вычислении `i`, `j` и `l` на каждой итерации. Также возникали проблемы с вычислением больших чисел и погрешностями. Для проверки я написал программу, использующую `math.h` и сверил вывод обеих программ. Приведенный ниже код не использует эту никакие библиотеки кроме стандартной ввода вывода данных и производит верные вычисления.

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

```
#include <stdio.h>
#define max(a, b) (((a) > (b)) ? (a) : (b))
#define min(a, b) (((a) < (b)) ? (a) : (b))

double abs(double a) {
    if (a >= 0) return a;
    else return -a;
}

double fmod(double a, double b) {
    return a - (int)(a/b) * b;
}

double pow2(double x) {
    return x * x;
}

int sgn(double a) {
    return a / abs(a);
}

int check(double i, double j) {
    if (pow2(i - 20) / 100 + pow2(j - 0) / 25 <= 1) return 1;
    else return 0;
}

void output (double i, double j, int k) {
    if (check(i, j)) {
        printf("k = %d: i = %f, j = %f:   HIT\n", k + 1, i, j);
    } else {
        printf("k = %d: i = %f, j = %f:   MISS\n", k + 1, i, j);
    }
}

int main() {
    double i = -24, j = 4, l = -3, ik = 0, jk = 0, lk = 0;
    for (int k = 0; k < 50; k++) {
        ik = i, jk = j, lk = l;
        i = fmod(abs((ik + k) * (jk + 2 * k) * (lk + 3 * k)), 35);
        j = sgn(max(ik, jk)) * min(fmod((ik + k), 20.0), fmod((jk + lk), 20.0));
        l = jk / 3 - abs(jk - k) * sgn(lk - jk);
        output(i, j, k);
    }
    return 0;
}
```

**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
0	дом	10.12.2022	18:00	Начал делать лабу, узнал, что нельзя юзать math.h	Терпение и труд все перетрут	Боль
1	дом	11.12.2022	15:00	Доделал Лабу	Сел писать отчет	Хахаха отдых
2	дом	11.12.2022	15:30	Отчет готов	Можно	Сдаваться

#### 10. Замечания автора по существу работы

Верните math.h

#### 11. Выводы

Си – это больно

Недочёты при выполнении задания могут быть устранены следующим образом: --

Подпись студента \_\_\_\_\_