

# Отчет по лабораторной работе № 9 по курсу «Фундаментальная информатика»

Студент группы Алапанова Эльза Халилевна, № по списку 3

Контакты e-mail : alapanowa02@yandex.ru

Работа выполнена: «20» ноября 2020г.

Преподаватель: каф. 806 Найденев Иван Евгеньевич

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

Подпись преподавателя \_\_\_\_\_

1. **Тема:** Системы программирования на языке Си.
2. **Цель работы:** Составление и отладка простейшей программы на языке Си итеративного характера с целочисленными рекуррентными соотношениями, задающими некоторое регулярное движение точки в целочисленной системе координат  $(i, j)$  с дискретным временем  $k$  и динамическим параметром  $l$ .
3. **Задание (вариант №29):** Необходимо определить попадание движущейся по двумерной целочисленной сетке точки в специальную геометрическую область не более чем за 50 шагов. Передвижение точки определяется специальными формулами и зависит от её текущего положения  $(i, j)$ , динамического параметра движения  $l$  и дискретного времени  $k$ . Положение точки на сетке  $(i_0, j_0)$  и значение динамического параметра  $l_0$  в нулевой момент времени ( $k = 0$ ) подаются со стандартного потока ввода. Результатом работы программы является информация о попадании точки в заданную геометрическую область. Вычисления останавливаются при попадании точки в заданную область или при превышении допустимого количества шагов.  
Геометрическая область: эллипс с центром в точке  $(20, 0)$  и проходящей через точки  $(10, 0)$ ,  $(30, 0)$ ,  $(20, 5)$  и  $(20, -5)$ . Формулы для вычисления координат и параметра см. в методических указаниях.

## 4. Оборудование (студенческое)

Процессор Intel® Core™ i5-10210 @ 1.60 GHz с ОП 8192 Мб, НМД 512 Уб. Монитор 1920 x 1080

## Программное обеспечение (студенческое):

Операционная система семейства Windows, наименование Windows 10 домашняя версия \_\_\_\_\_  
интерпретатор команд \_\_\_\_\_ версия \_\_\_\_\_

Система программирования \_\_\_\_\_ версия \_\_\_\_\_

Редактор текстов \_\_\_\_\_ версия \_\_\_\_\_

Утилиты операционной системы \_\_\_\_\_

Прикладные системы и программы Sublime Text

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

Пока кол-во итераций меньше 50 и точка не попала в нужную зону - изменяем координаты по данным формулам. Если кол-во итераций стало 50 или точка попала в нужную область - программа завершается и выводится ответ.

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

План работы:

1) Прописываем функции `max`, `min`, `chek`, `mod` и `sign`.

2) Вводим  $i, j, l$ , выписываем для них формулы

3) Если попал в область эллипса - выводим да и числа. Если не попал в область или кол-во больше 50, то выводим нет.

Пункты 1-7 отчета составляются строго до начала лабораторной работы.

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

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#define n 50
```

```
int max(int x, int y)
{
    return (x > y) ? x : y;
}
```

```
int min(int x, int y)
{
    return (x > y) ? y : x;
}
```

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

```
int mod(int e, int d)
{
    return (d + e % d) % d;
}
```

```
int chek(int i, int j)
{
    if ((i - 20) * (i - 20) + 4 * j * j <= 100) {
        return 1;
    }
    return 0;
}
```

```
int main(void)
{
    int i, j, l;
    int i2, j2, l2;
    int k = 0;
```

```
    scanf("%d%d%d", &i, &j, &l);
```

```
    for (k = 0; k < n && chek(i, j) == 0; k++) {
        i2 = mod((max(max((i * j), (i * l)), (j * l))), 30) + k;
        j2 = abs(j - l) * sign(i) - abs(i - l) * sign(j);
        l2 = min(i, max(j, min(l, max((i - l), (j - l)))));
        i = i2;
        j = j2;
        l = l2;
    }
```

```
    if (chek(i, j)) {
        printf("Yes\n");
        printf("%d %d %d %d\n", i, j, l, k);
    } else {
        printf("No\n");
        printf("%d %d %d %d\n", i, j, l, n);
    }
    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
1						
2						

**10. Замечания автора** по существу работы : замечаний нет.

**11. Выводы :** работа не понравилась. Вроде нужно чисто подставить, но нужно еще догадаться, как подставлять формулу эллипса. Траектория попадания неясна, ничего не было понятно.

Недочёты при выполнении задания могут быть устранены следующим образом: \_\_\_\_\_

Подпись студента \_\_\_\_\_