



Отчёт по лабораторной работе № 9 по курсу 1

Студент группы 106 Мигалев Р. П., № по списку 11

Контакты www, e-mail, icq, skype migalev-roman@ya.ru

Работа выполнена: « 7 » ноября 201 6 г.

Преподаватель: ассист. каф.806 Дубинин А.В.

Входной контроль знаний с оценкой

Отчёт сдан « » 201 г., итоговая оценка

Подпись преподавателя

1. Тема: Движение точки

2. Цель работы: Составить и отладить программу 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 №23): Треугольник с вершинами в точках (-10,0), (0, 10), (-10, 20).
 $i_0 = 29, j_0 = -6, l_0 = 1,$
 $ik+1 = \min(\max(\min(ik - jk, ik - lk), jk - lk), ik - k) \bmod 30,$
 $jk+1 = \max(\min(\max(ik - jk, ik - lk), jk - lk), ik - k) \bmod 30,$
 $lk+1 = ik \bmod 30 - jk \bmod 30 + lk \bmod 30 - k \bmod 30.$

4. Оборудование(лабораторное):

ЭВМ, процессор, имя узла сети с ОП Мб,
НМД Мб. Терминал адрес. Принтер
Другие устройства

Оборудование ПЭВМ студента, если использовалось:

Процессор Intel Core i7 4510U с ОП 8192 Мб, НМД 1048576 Мб. Монитор
Другие устройства

5. Программное обеспечение(лабораторное):

Операционная система семейства, наименование версия
интерпретатор команд версия
Система программирования версия
Редактор текстов версия
Утилиты операционной системы

Прикладные системы и программы
Местонахождение и имена файлов программ и данных

Программное обеспечение ЭВМ студента, если использовалось:

Операционная система семейства Linux, наименование Ubuntu (Xenial Xerus) версия 16.04 LTS
интерпретатор команд GNU bash версия 4.3.46(1).
Система программирования версия
Редактор текстов VIM - Vi IMproved версия 7.4
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)

Реализовать на языке Си программу, итеративно моделирующую движение точки на координатной плоскости в зависимости от заданного алгоритма ее передвижения. Для корректного решения данной задачи также необходимо написать остаток от деления по Евклиду.

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

Создать цикл на 50 итераций и обновлять текущие параметры точки. Если она достигла заданной области, вывести сообщение об этом, указав количество шагов (текущую итерацию). Иначе после выполнения цикла вывести сообщение о превышении допустимого числа шагов, сопроводив сообщение текущими параметрами точки.

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).

```
#include <stdio.h>
#include <math.h>

#define min(x, y) (((x) < (y)) ? (x) : (y))
#define max(x, y) (((x) > (y)) ? (x) : (y))
#define abs(x) (((x) > 0) ? (x) : (-x))

int mod(int x, int y)
{
    int r = x % y;

    if (r > 0)
        return r;
    else
        return r + abs(y);
}

int main()
{
    int icur = 29;
    int jcur = -6;
    int lcur = 1;

    int iprev, jprev, lprev;

    // A(-10, 0) B(0, 10) C(-10, 20)
    int x[3] = {-10, 0, -10};
    int y[3] = {0, 10, 20};

    for (int k = 0; k < 50; k++)
    {
        int a = (x[1] - icur) * (y[2] - y[1]) - (x[2] - x[1]) * (y[1] - jcur);
        int b = (x[2] - icur) * (y[3] - y[2]) - (x[3] - x[2]) * (y[2] - jcur);
        int c = (x[3] - icur) * (y[1] - y[3]) - (x[1] - x[3]) * (y[3] - jcur);

        if (((a >= 0) && (b >= 0) && (c >= 0)) || ((a <= 0) && (b <= 0) && (c <= 0)))
        {
            printf("Specified area reached at point (%d, %d) in %d steps.\n", icur, jcur, k);

            return 0;
        }

        iprev = icur;
        jprev = jcur;
        lprev = lcur;

        icur = mod(min(max(min(iprev - jprev, iprev - lprev), jprev - lprev), iprev - k), 30);
        jcur = mod(max(min(max(iprev - jprev, iprev - lprev), jprev - lprev), iprev - k), 30);
        lcur = mod(iprev, 30) - mod(jprev, 30) + mod(lcur, 30) - mod(k, 30);
    }

    printf("Exceeded steps limit, stopping at point (%d, %d). Current l = %d.\n", icur, jcur, lcur);

    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

10. **Замечания автора** по существу работы _____

11. Выводы

В результате проведения лабораторной работы было успешно освоено написание программ итеративного характера на примере движения точки.

Недочёты при выполнении задания могут быть устранены следующим образом: _____

Подпись студента _____