



Отчёт по лабораторной работе № 9 по курсу 1 Фундаментальная информатика

Студент группы М8О-104Б-22 Алиев Руслан Магомедович, № по списку: 1

Контакты www, e-mail, icq, skype zfacky@mail.ru

Работа выполнена: “22 “ октября 2022г.

Преподаватель: каф. 804 Потенко М.А

Входной контроль знаний с оценкой _____

Отчёт сдан “___ “ _____ 20___ г., итоговая оценка _____

Подпись преподавателя _____

- 1 **Тема:** Программирование на языке С.
- 2 **Цель работы:** Составление и отладка простейшей программы на языке Си итеративного характера с целочисленными рекуррентными соотношениями, задающими регулярное некоторое движение точки в целочисленной системе координат (i, j) с дискретным временем k и динамическим параметром движения l.
- 3 **Задание (вариант № 24):**

V. Треугольник с вершинами в точках $(-10, 0)$, $(0, 10)$, $(-10, 20)$

$$\begin{aligned} 24. \quad & i_0 = 20, j_0 = 0, l_0 = 11 \\ & i_{k+1} = ((i_k - k) \max(j_k, l_k) + (j_k - k) \min(i_k, l_k) + (l_k - k) \max(i_k, j_k)) \bmod 23, \\ & j_{k+1} = -((i_k - k) \min(j_k, l_k) + (j_k - k) \max(i_k, l_k) + (l_k - k) \min(i_k, j_k)) \bmod 27, \\ & l_{k+1} = |i_k + j_k - l_k - k| \operatorname{sign}(i_k - j_k + l_k - k) \end{aligned}$$

- 4 **Оборудование ПЭВМ студента :** Процессор Apple M1 с ОП 8000Мб, НМД 257024 Мб. Монитор Retina
Другие устройства не использовались
- 5 **Программное обеспечение ПЭВМ студента (лабораторное):**
Операционная система семейства macOS, наименование: macOS Monterey версия 12.5.1
Интерпретатор команд: zsh версия 5.8.1
Редактор текстов: nano
Утилиты операционной системы: Терминал
Прикладные системы и программы: Safari
Местонахождения и имена файлов программ и данных: /Users/yoonseak/Desktop
- 6 **Идея, метод, алгоритм** решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальное описание с пред- и постусловиями):

Самая первая идея, которая возникла – запоминать i, j, l на каждой итерации добавляя их в массив. Придерживаясь именно этой идеи выполнял задание (не самое верное решение, так как уходит много памяти на выделение 3-х массивов по 50 чисел, но для решения данной задачи не слишком критично). Затем, второй раз пробегаясь по массивам, подставляя значения и производить проверку: попадает ли точка в заданную область или нет. Если точка попадет, то выводить соответствующий текст, иначе выводить, что ни на одной из итераций точка не попала в заданную область. Сложность алгоритма “О-большое” получается равной $O(n^2)$. Данный алгоритм не является эффективным, но для задачи он будет выполняться не так долго, так как имеем относительно не большие массивы чисел.
- 7 **Сценарий выполнения работы** [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты либо соображения по тестированию].

Сначала подключил библиотеки при помощи `#include` ввода и вывода, математических функций. Затем задал константы при помощи метода `#define`. Следующим шагом написал, вне основного тела функции, функции:

max (выбор максимального из пары чисел), sign (определение знака числа), abs (взятие числа по модулю), min (выбор минимального из пары чисел). Затем создаю массивы и присваивая первым элементам массива соответствующие начальные I, j, l. Завожу цикл на количество итераций равное KMAX (50) и вычисляю k+1 элемент массива для I, j, l. Следующим шагом применяю векторное и псевдоскалярное произведение, которому свойственно следующее: Если они одинакового знака, то точка внутри треугольника, если что-то из этого - ноль, то точка лежит на стороне, иначе точка вне треугольника. Если условие выполняется, то вывожу координаты точки и итерацию, на которой это условие выполнено. Иначе, если ни на одной итерации цикла ни одна точка не попала в треугольник, то вывожу, что ни одна из точек не попала в треугольник.

Допущен к выполнению работы. **Подпись преподавателя** _____

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

```
//подключение библиотек ввода и вывода математических функций
#include <stdio.h>
#include <math.h>

//Задаём константные переменные, а именно координаты вершин треугольника
#define X1 -10
#define Y1 0
#define X2 0
#define Y2 10
#define X3 -10
#define Y3 20
#define KMAX 50

//функция максимума из 2 чисел
int max(int a, int b)
{
    if (a >= b) return a;
    if (a < b) return b;
}

//функция которая определяет знак числа
int sign(a)
{
    if (a > 0) return 1;
    if (a < 0) return -1;
    if (a == 0) return 0;
}

//функция модуля
int abs(a)
{
    if (a >= 0) return a;
    if (a < 0) return -a;
}

//функция минимума из 2 чисел
int min(int a, int b)
{
    if (a >= b) return b;
    if (a < b) return a;
}

//основная функции
int main()
{
    //Ввожу счетчик точек, которые попадут в треугольник после выполнения программы
    int count = 0;
```

```

//Создаю три массива
int i[50];
int j[50];
int l[50];
//Присваиваю первому элементу массива значение i,j,l первых
i[0] = 20;
j[0] = 0;
l[0] = 11;
//Цикл
for (int k = 0; k <= KMAX; ++k) {
    //Вычисление элемента под индексом k+1 для i,j,l
    i[k + 1] = ((i[k] - k) * max(j[k], l[k]) + (j[k] - k) * min(i[k], l[k]) + (l[k] - k) * max(i[k], l[k])) % 23;
    j[k + 1] = -(i[k] - k) * min(j[k], l[k]) + (j[k] - k) * max(i[k], l[k]) + (l[k] - k) * min(i[k], l[k])) % 27;
    l[k + 1] = abs(i[k] + j[k] - l[k] - k) * sign(i[k] - j[k] + l[k] - k);

    //Применяю векторное и псевдоскалярное произведение,которому свойственно
    следующее:Если они одинакового знака,то точка внутри треугольника,если что-то из этого - ноль,то
    точка лежит на стороне,иначе точка вне треугольника
    if (((X1 - i[k]) * (Y2 - Y1) - (X2 - X1) * (Y1 - j[k]) >= 0) &&
        ((X2 - i[k]) * (Y3 - Y2) - (X3 - X2) * (Y2 - j[k]) >= 0) &&
        ((X3 - i[k]) * (Y1 - Y3) - (X1 - X3) * (Y3 - j[k]) >= 0)) || (((X1 - i[k]) * (Y2 - Y1) - (X2 - X1) *
(Y1 - j[k]) < 0) &&
        ((X2 - i[k]) * (Y3 - Y2) - (X3 - X2) * (Y2 - j[k]) < 0) && ((X3 - i[k]) * (Y1 - Y3) - (X1 - X3) * (Y3
- j[k]) < 0))) {

        count += 1;
        printf("номер точки=%d\n", count);
        printf("x=%d\n", i[k]);
        printf("y=%d\n", j[k]);
        printf("дискретное время=%d\n", k);
    }

}

//Пишу return,так как функция типа "int" должна что-либо возвращать
return 0;
}

```

```

yoonseak@MacBook-Air-Ruslan Срап % gcc main7.c
main7.c:20:1: warning: non-void function does not return a value in all control paths [-Wreturn-type]
}
^
main7.c:28:1: warning: non-void function does not return a value in all control paths [-Wreturn-type]
}
^
main7.c:34:1: warning: non-void function does not return a value in all control paths [-Wreturn-type]
}
^
main7.c:40:1: warning: non-void function does not return a value in all control paths [-Wreturn-type]
}
^
4 warnings generated.
yoonseak@MacBook-Air-Ruslan Срап % ./a.out
номер точки=1
x=-9
y=6
дискретное время=20
номер точки=2
x=-9
y=14
дискретное время=45
zsh: abort ./a.out

```

- 9 **Дневник отладки** должен содержать дату и время сеансов отладки, и основные ошибки (ошибки в сценарии и программе, не стандартные операции) 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 **Замечание автора по существу работы:**

11 **Выводы**

При написании первой серьезной программы я научился правильно работать с языком программирования Си. Я искал информацию о командах на форумах, учился правильно писать код (правильно оформлять) и учил синтаксис. Смог хорошо справиться со своим заданием и сделать первый серьезный шаг к становлению программистом. Написание этой программы замотивировало изучать этот язык ещё более углублено. С интересом выполнять лабораторные работы и заниматься ими. Такие работы учат нас базовым алгоритмам программирования на Си, которые в будущем нам помогут писать более сложные программы

Недочёты при выполнении задания могут быть устранены следующим образом:

Подпись студента _____