

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

Студент группы М80-109Б-22 Моравская Варвара Ивановна, № по списку 11

Контакты e-mail, telegram, skype @Vavar1st

Работа выполнена: «19» ноября 2022г.

Преподаватель: каф. 806 Сысоев Максим Алексеевич

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 С итеративного характера с целочисленными рекуррентными соотношениями.
3. **Задание:** Вариант 23.

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

$$\begin{aligned} 23. \quad & i_0 = 29, j_0 = -6, l_0 = 1 \\ & i_{k+1} = \min(\max(\min(i_k - j_k, i_k - l_k), j_k - l_k), i_k - k) \bmod 30, \\ & j_{k+1} = \max(\min(\max(i_k - j_k, i_k - l_k), j_k - l_k), i_k - k) \bmod 30, \\ & l_{k+1} = i_k \bmod 30 - j_k \bmod 30 + l_k \bmod 30 - k \bmod 30 \end{aligned}$$

4. **Оборудование** (студента):
Процессор *Intel(R) Core(TM) i5-2410M CPU @ 2.30GHz* 2.30 GHz ОП 8 зб, 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 *nano* версия *2.5.2.2*
Утилиты операционной системы --
Прикладные системы и программы --
Местонахождение и имена файлов программ и данных на домашнем компьютере --

6. Идея, метод, алгоритм:

Для начала я вручную прописала функции `min`, `max`, `mod`. Также я прописала функции изменения каждого из параметров. Далее я воспользовалась одним из алгоритмов на определение нахождения точки внутри треугольника, а именно ориентационный подход. Его суть заключается в том, чтобы определить с какой стороны (справа или слева) находится точка относительно каждой из сторон треугольника. В этом случае мы представляем стороны, как векторы, соединенные последовательно. И если для каждого вектора точка окажется с одной и той же стороны, значит точка находится в треугольнике. Собственно, для этого сначала создаем функцию, которая будет высчитывать координаты векторов и определять, с какой стороны окажется точка. С одной стороны – возвращаем 1, с другой – 0. Далее создаем функцию окончательного определения точки относительно всех сторон треугольника. Тут указаны координаты точек и подсчитывается сумма «сторон», на которых была точка. Т.к. нам нужно, чтобы точка была с одной стороны для каждого из векторов, нас устроит три 1 или три 0, следовательно, 3 или 0. Наконец, мы подходим к самой программе. Задаем начальные значения из условия и запускаем `while`. Нас интересуют до 50 операций, поэтому, если $k < 50$, мы изменяем каждый из параметров согласно условию. Если обнаружится, что мы попали в треугольник, выведется «Hit» и все параметры, иначе – «Miss» и все параметры.
По результатам моего варианта, в течении первых 50 шагов точка НЕ попадет в треугольник.

7. Сценарий выполнения работы

Результат:

Miss. $k = 50$ $i = 23$ $j = 4$ $l = 11$

8. Распечатка протокола

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

```
int min(int a, int b) {
```

```

    if (a < b) {
        return a;
    }
    else {
        return b;
    }
}

```

```

int max(int a, int b) {

```

```

    if (a > b) {
        return a;
    }
    else {
        return b;
    }
}

```

```

int mod(int N, int n) {

```

```

    if (N < 0) {
        N = N * (-1);
    }
    if (n < 0) {
        n = n * (-1);
    }
    while (n < N) {
        N -= n;
    }
    return N;
}

```

```

int count_i(int i, int j, int l, int k) {
    int ik = mod(min(max(min((i - j), (i - l)), (j - l)), (i - k)), 30);
    return ik;
}

```

```

int count_j(int i, int j, int l, int k) {
    int jk = mod(max(min(max((i - j), (i - l)), (j - l)), (i - k)), 30);
    return jk;
}

```

```

int count_l(int i, int j, int l, int k) {
    int lk = (mod(i, 30)) - (mod(j, 30)) + (mod(l, 30)) - (mod(k, 30));
    return lk;
}

```

```

int Orientation_calc(int P1_x, int P1_y, int P2_x, int P2_y, int P3_x, int P3_y) {
    int vector_1_x = P2_x - P1_x;
    int vector_2_x = P3_x - P1_x;

```

```

    int vector_1_y = P2_y - P1_y;
    int vector_2_y = P3_y - P1_y;
    int cross_product = vector_1_x * vector_2_y - vector_1_y * vector_2_x;
    if (cross_product > 0) {
        return 1;
    }
    else {
        return 0;
    }
}

```

```

int inside_triangle(int P_x, int P_y) {

```

```

const int A_x = -10;
const int A_y = 0;
const int B_x = 0;
const int B_y = 10;
const int C_x = -10;
const int C_y = 20;

int turns = Orientation_calc(A_x, A_y, B_x, B_y, P_x, P_y) + Orientation_calc(B_x, B_y, C_x, C_y, P_x, P_y) +
Orientation_calc(C_x, C_y, A_x, A_y, P_x, P_y);
if ((turns == 3)|| (turns == 0)) {
    return 1;
}
else {
    return 0;
}

}

int main() {

    int check = 0;
    int i = 29;
    int j = -6;
    int l = 1;
    int k = 0;

    while (k < 50) {
        if (inside_triangle(i, j)) {
            printf("Hit. k = %d\t i = %d\t j = %d\t l = %d\n", k, i, j, l);
            check = 1;
            break;
        }
        else {
            int ik = count_i(i, j, l, k);
            int jk = count_j(i, j, l, k);
            int lk = count_l(i, j, l, k);
            k += 1;

            i = ik;
            j = jk;
            l = lk;
        }
    }
    if (check == 0) {
        printf("Miss. k = %d\t i = %d\t j = %d\t l = %d\n", k, i, j, l);
    }

    return 0;

}

```

9. Дневник отладки

№	Лаб.	Дата	Время	Событие	Действие по исправлению	Примечание
1	дом	18.11.2022	вечер	Ничего не работало	Исправила синтаксис	Все равно не работает :/
2	дом	19.11.2022	день	Ничего не выводится	Снова исправила синтаксис	Заработало ☺

10. Замечания автора

Спотыкаясь о незнакомый мне синтаксис, я пыталась написать эту программу. Мне даже пришлось прошерстить интернет в поиске алгоритмов для выполнения работы. Было трудно, но, вроде, я справилась.

11. Выводы

Большинство вопросов, возникающих у меня на этапе написания программы, мне удавалось решить, почитав документацию по Си. Все-таки отсутствие опыта программирования на данном языке – самый главный мой враг. Но интернет, конечно, в помощь. Поэтому не стесняемся гуглить ☺.

Недочёты при выполнении задания могут быть устранены следующим образом: -- изучаем синтаксис (опять)!!!

Подпись студента _____