

Отчет по лабораторной работе № 9 по курсу Фундаментальная информатика

Студент группы М8О-104Б-22 Ляпин Иван Алексеевич, № по списку 00

Контакты www, e-mail, icq, skype shad0w2020@mail.ru

Работа выполнена: « 14 » октября 2022 г.

Преподаватель: асп. каф. 806 Потенко М.А.

Входной контроль знаний с оценкой _____

Отчет сдан « » _____ 202 __ 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, задающими некоторое регулярное движение точки в целочисленной системе координат (i, j) с дискретным значением времени k и динамическим параметром движения l . _____

3. **Задание (вариант № 4):** Кольцо, ограниченное двумя окружностями с центром в точке $(10, 10)$, радиус внутренней окружности равен 5, а радиус внешней равен 10. _____

4. **Оборудование (лабораторное):**
ЭВМ _____, процессор _____, имя узла сети _____ с ОП _____ Мб,
НМД _____ Мб. Терминал _____ адрес _____. Принтер _____
Другие устройства _____

Оборудование ПЭВМ студента, если использовалось:

Процессор Apple M1 Pro с ОП 16384 Мб, НМД 524 288 Мб. Монитор Liquid Retina XDR

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 macOS, наименование macOS Monterey версия 12.3

интерпретатор команд zsh версия 2.12.5

Система программирования С версия _____

Редактор текстов nano версия _____

Утилиты операционной системы Терминал _____

Прикладные системы и программы Xcode

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

Напишем код, благодаря которому сможем вычислять параметры переменных по формулам, приложенным в задании.

Пропишем самостоятельно функцию `sign`, а также импортируем библиотеку `<math.h>` для дальнейших вычислений.

В главной функции объявим входные данные переменных, а также будущие переменные.

Затем составим цикл от $k=0$ до $k=50$. Очевидно что, если наша точка не смогла оказаться в указанной области за 50 шагов, => выводим промах и координаты с динамическим параметром движения.

Заметим что, точка может принадлежать указанной области тогда и только тогда, когда выполняется следующая система:

$(i0-ICENTER)*(i0-ICENTER)+(j0-JCENTER)*(j0-JCENTER))>=25$ $(i0-ICENTER)*(i0-ICENTER)+(j0-JCENTER)*(j0-JCENTER)<=100$, где $j0, i0$ - изменяемые значения входных переменных.

Если точка попадёт в указанную область, то программа выведет координаты попадания, а также номер шага и значение динамического параметра движения, на котором это произошло.

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

1) Создадим файл в утилите "Xcode" для написания кода программы.

2) Далее сохраняем данный код файлом в формате .c.

3) При помощи терминала переходим в директорию с этим файлом.

4) Командой "cat" выведем его содержимое(то есть сам код).

5) Скомпилируем файл при помощи компилятора gcc и одноимённой команды.

6) С помощью команды "./a.out" получаем возможность работы с написанной нами командой, вводим 2 произвольных числа и выводим результат.

Допущен к выполнению работы. Подпись преподавателя _____

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

Код из файла 9labb.c

```
/*
    Лапин Иван Алексеевич, И80-1046-22, вариант №4.
*/
#include <stdio.h>
#include <math.h>

#define I 26 // i - начальная координата
#define J 8 // j - начальная координата
#define L -3 // l - начальное значение
#define R1 25 // Квадрат радиуса малой окружности
#define R2 100 // Квадрат радиуса большей окружности
#define ICENTER 10 // i - координата центра окружностей
#define JCENTER 10 // j - координата центра окружностей
#define NUMBER_STEPS 50 // номер последнего шага счётчика k

/* Пишем функцию sign */
int sign (int a)
{
    if (a > 0) {
        return 1;
    } else if (a < 0) {
        return -1;
    } else {
        return 0;
    }
}

int main (int argc, const char * argv[])
{
    /* Объявляем переменные, принимающие результат и присваиваем им результат */
    int i_1 = I, j_1 = J, l_1 = L, k = 0, i, j, l;
    /* Входим в цикл, где будет происходить перебор k от 0 до 50 на проверку попадания*/

    for (k = 0; k < NUMBER_STEPS; k++) {
        /* Пусть i, j - конечные значения исходных i_1, j_1, после операций над ними */

        i = i_1;

        j = j_1;

        l = l_1;

        /* формулы, что были даны в варианте */
        i_1 = fmod(fmin(i + j, i + 1), 30);

        j_1 = j + fmod(1 * sign(j), 20) + fmod(k * sign(i), 10);

        l_1 = fmod(fmax(fmax(i * j, i * 1), j * 1), 30);

        /* Пишем условие вхождения в указанную область. Причем очевидно, что наша точка будет принадлежать заданной окрестности, только если будет принадлежать большей окружности, но при этом не будет принадлежать меньшей */
        if (((((i_1 - ICENTER) * (i_1 - ICENTER) + (j_1 - JCENTER) * (j_1 - JCENTER)) >= R1) && (((i_1 - ICENTER) * (i_1 - ICENTER) * (j_1 - JCENTER) * (j_1 - JCENTER) <= R2))) {
            /* Выводим номер шага, в котором мы попали и соответствующие координаты, параметр движения */
            printf("Попадание! Шаг № %d. \n", k);
            printf("i = %d, j = %d, l = %d.\n", i_1, j_1, l_1);
            /* Выход из цикла */
            return 0;
        }

    }

    /* Если точка так и не попала в цикл в заданную область, то выводим координаты промаха с дополнительной информацией */
    printf("i = %d, j = %d, l = %d.\n", i_1, j_1, l_1);
    /* Выводим все необходимые параметры промаха(i_1,j_1,l_1) */
    printf("Промак!Координаты: (%d,%d). Динамический параметр движения: %d. Шаг №50.\n", i_1, j_1, l_1);

    return 0;
}
```

Вывод терминала:

```
Last login: Thu Oct 27 21:01:07 on console
ivan@MacBook-Pro-Ivan ~ % cd Desktop
ivan@MacBook-Pro-Ivan Desktop % cat 9labb.c
/* код */
ivan@MacBook-Pro-Ivan Desktop % gcc 9labb.c
2022-10-27 21:03:37.116 xcodebuild[673:7452] Requested but did not find extension point with identifier Xcode.IDEKit.ExtensionSentinelHostApplications for extension Xcode.DebuggerFoundation.AppExtensionHosts.watchOS of plug-in com.apple.dt.IDEWatchSupportCore
2022-10-27 21:03:37.116 xcodebuild[673:7452] Requested but did not find extension point with identifier Xcode.IDEKit.ExtensionPointIdentifierToBundleIdentifier for extension Xcode.DebuggerFoundation.AppExtensionToBundleIdentifierMap.watchOS of plug-in com.apple.dt.IDEWatchSupportCore
ivan@MacBook-Pro-Ivan Desktop % ./a.out 9labb.c
i = 0, j = 198, l = 0 .
Промак!Координаты: (0,198). Динамический параметр движения: 0. Шаг №50.
ivan@MacBook-Pro-Ivan Desktop %
```

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 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: Благодаря данной лабораторной работе я научился составлять простейшие программы на языке C, а также научился работать с компилятором gcc в терминале и получил необходимые навыки при работе с текстами и отладкой программы. Стоит отметить, что в данной лабораторной работе были задействована библиотека `<math.h>`, благодаря которой я сумел вычислить искомые параметры, также был задействован цикл `for` и были применены условные конструкции(`if`).

Недочёты при выполнении задания могут быть устранены следующим образом:

Подпись студента

A handwritten signature in black ink, consisting of stylized, overlapping loops and strokes, positioned on a horizontal line.