

# Отчет по лабораторной работе № 9 по курсу Архитектура компьютера и информационных сетей

Студент группы М8О-103Б-22 Клименко Виталий Максимович, № по списку 11

Контакты www, e-mail, icq, skype vitalikklimenko96@gmail.com

Работа выполнена: 28 октября 2022 г.

Преподаватель: доцент Никулин С.П.

Входной контроль знаний с оценкой \_\_\_\_\_

Отчет сдан «    » \_\_\_\_\_ 202 \_\_ г., итоговая оценка \_\_\_\_

Подпись преподавателя \_\_\_\_\_

1. **Тема:** Простейшая программа на Си
2. **Цель работы:** Составление и отладка простейшей программы на Си интерактивного характера
3. **Задание (вариант № 13):** Лунка, являющаяся пересечением двух кругов радиуса 10, центр первого круга - в точке (-10, -10), центр второго - в точке (-20, -20)
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 4x 3.5GHz с ОП 16 ГБ НМД HDD 200 ГБ . Монитор Встроенный 1920x1080  
Другие устройства Touchpad Synaptics
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 UNIX, наименование Pop!\_OS версия 22.04 jammy  
интерпретатор команд bash версия 5.1.16  
Система программирования GNU Compiler Collection (GCC) версия 11.2.0  
Редактор текстов Visual Studio Code версия 1.72.2  
Утилиты операционной системы \_\_\_\_\_  
Прикладные системы и программы gcc  
Местонахождение и имена файлов программ и данных на домашнем компьютере \_\_\_\_\_

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

Написать математические функции `sign`, `abs` и т.д. для алгоритма, создать список с размером из условия, в котором будут записываться все значения координат и дополнительной переменной `l`. Просчитывать новые переменные до тех пор, пока не будет ясен результат

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

1. Написать математические функции
2. Запустить цикл, в котором будут считаться следующие значения
3. Вывести все значащие переменные если точка попала в заданную область, иначе сообщить о непадании

*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)

```
vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/19$ cat main.c
/* Лабораторная работа №9, вариант №13
 * Студент гр. М80-103Б-22 Клименко В. М. */
#include<stdio.h>

short calc(long long i, long long j) {
    if ((i + 10)*(i + 10) + (j + 10)*(j + 10) <= 100
        && (i + 20)*(i + 20) + (j + 20)*(j + 20) <= 100) {
        return 1;
    }
    return 0;
}

long long max(long long a, long long b) {
    if (a > b) { return a; }
    return b;
}

long long min(long long a, long long b) {
    if (a < b) { return a; }
    return b;
}

long long abs(long long a) {
    if (a < 0) { return -a; }
    return a;
}

long long sign(long long a) {
    if (a > 0) { return 1; }
    if (a == 0) { return 0; }
    return -1;
}

long long mod(long long a, long long b) {
    long long fl = floor((double) a / (double) b);
    return a - fl * b;
}

long long floor(double x) {
    if (x - (long long) x >= 0) {
        return (long long) x;
    }
    return (long long) x - 1;
}

int main() {
    /*! for writing i, j, k for graphics
    FILE *fp;
    fp = fopen("for_graphics.txt", "w");

    long long ip = 13, jp = -9, lp = -4, ii, jj, ll;
    int kk;
    short gg = 0;

    for (int k = 1; k < 50; ++k) {
        kk = k - 1;

        ii = ip;
        jj = jp;
        ll = lp;

        ip = mod(ii + jj, 30) / (mod(abs(ll), 5) + 1) + mod(ii + ll, 30)/(mod(abs(jj), 5) + 1);
        jp = mod(max(kk * ii, (kk + 1) * jj), 25) - abs(jj - ll)/10;
        lp = abs(jj - ll)/10 + min(mod(ii + ll, 20), mod(jj * kk, 20)) - 10;
        fprintf(fp, "k = %d:\ni = %lld\nj = %lld\nl = %lld\n", k, ip, jp, lp);

        if (calc(ip, jp) == 1) {
            printf("Point is in at k = %d:\ni = %lld\nj = %lld\nl = %lld\n", k, ip, jp, lp);
            gg = 1;
            break;
        }
    }

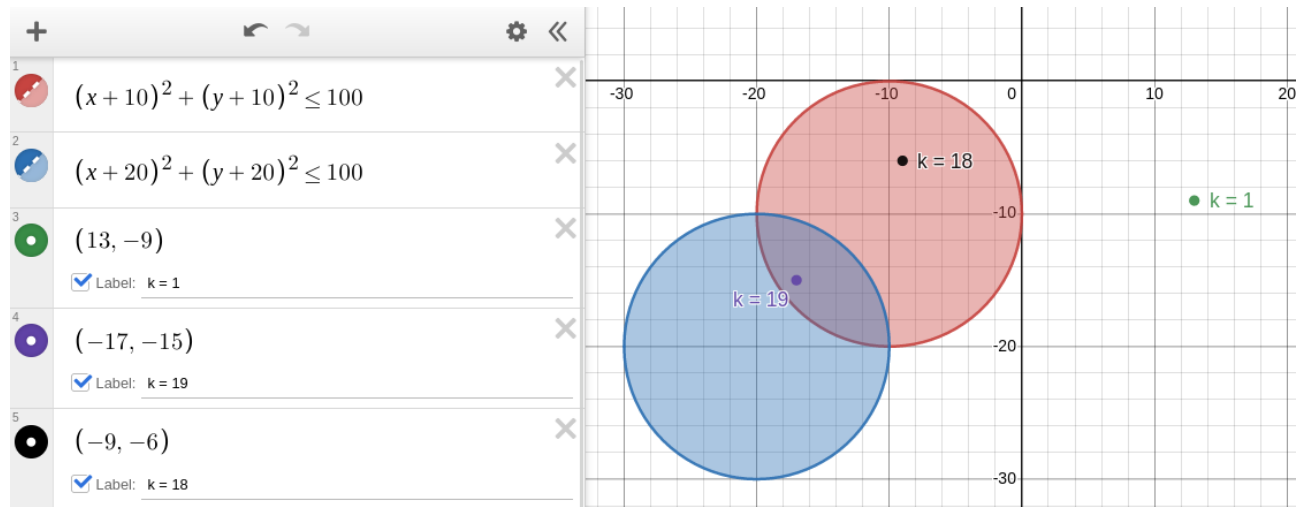
    if (gg == 0) {
        printf("Couldn't make it in 50 iterations! :(\n");
    }

    fclose(fp);
```

```

    return 0;
}vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/19$ gcc main.c -o main
vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/19$ ./main
Point is in at k = 19!
i = -17
j = -15
l = -17

```



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:** Я научился писать программы низкой сложности на Си, больше узнал про функции, ввод-вывод. Научился открывать файлы с помощью Си и записывать в них информацию. Узнал больше про ошибки выполнения и про то, что очень часто Си может не показывать, что происходит ошибка с типами.

Недочёты при выполнении задания могут быть устранены следующим образом: Можно было не хранить массив для всех значений i, j, l, а просто сохранять последнее для уменьшения необходимой памяти для программы.

Подпись студента \_\_\_\_\_