



Отчет по лабораторной работе № 09 по курсу Фундаментальная информатика

Студент группы М8О-104Б-22 Жарков Александр Алексеевич, № по списку 4

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

Работа выполнена: « 19 » 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:** Программирование на языке Си

2. **Цель работы:** Составление и отладка простейшей программы на языке Си с целочисленными рекуррентными соотношениями, задающими регулярное движение точки в целочисленной системе координат (i,j), дискретным временем k и динамическим параметром движения l.

3. **Задание (вариант № 7):** Квадрат с длиной стороны 10, стороны квадрата параллельны осям координат, центр квадрата в точке (10, -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 AMD RYZEN 5 5600H с ОП 16384 Мб, ssd 524288 Мб. Монитор IPS 1920*1080

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 UNIX, наименование Ubuntu версия 20.04 LTS

интерпретатор команд bash версия 5.1.16

Система программирования _____ версия _____

Редактор текстов gedit версия 3.30.2

Утилиты операционной системы Терминал

Прикладные системы и программы _____

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)

Обозначим необходимые значения константами, напомним вспомогательные математические функции и функции изменения координат, функцию проверки попадания в заданную область. Запустим цикл, на каждом шаге будем изменять параметры движения и проверять, попали ли мы в заданную область или нет. Если да-выведем значения параметров и номер шага, в который случилось попадание. Если нет-пропустим этот шаг и будем проверять попадание на следующем шаге. После 50 шагов закончим выполнение программы. Если за все 50 шагов не случилось ни одного попадания-выведем значения параметров на последнем шаге и закончим выполнение программы.

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

Подключим нужные для написания программы библиотеки: `stdio.h`, `math.h`, `stdlib.h`, `stdbool.h`. Создадим константами с помощью `define` начальные параметры движения, максимальное количество шагов, размеры квадрата и координаты его центра. Создадим функцию `mod` (нахождение остатка от деления), функции поиска минимума и максимума из двух чисел, функции динамического (пошагового) изменения параметров движения (координат i и j , параметра движения k), изменения даны изначально (берем из задания). Также создадим функцию проверки попадания в заданную область (квадрат). Если каждая из координаты движения (точка) попала внутрь квадрата (лежит правее/выше самой левой/нижней точкой квадрата (которая вычисляется как координата центра квадрата минус половина длины его стороны) и при этом левее/ниже самой правой/высокой точки квадрата (которая вычисляется как координата центра квадрата плюс половина длины его стороны)), то функция вернет 1, иначе она вернёт 0. Создадим счётчик попаданий в заданную область и запустим цикл шагов по k (с 0 до максимального возможного $k = 50$), изменяя на каждом шаге параметры движения и проверяя попадание в заданную область. При попадании увеличим счётчик попаданий и выведем значение каждого параметра движения и номер шага, на котором случилось попадание. После прохождения проверим счётчик попаданий. Если он равен 0 (ни одного попадания не случилось), то выведем сообщение о том, что количество шагов исчерпано и значения параметров движения на последнем шаге. После этого закончим выполнение программы.

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)

```
1#include <stdio.h> //Жарков Александр Алексеевич
2#include <math.h> //Лабораторная работа 9
3#include <stdlib.h> //Вариант 7
4#include <stdbool.h>
5
6#define I0 11 //Объявляем константы - заданные значения
7#define J0 13
8#define L0 10
9#define K_MAX 50
10#define CENTRE_I 10
11#define CENTRE_J -10
12#define SQUARE_SIDE 10
13
14
15
16int mod(int a, int b) { //Функция mod (взятие остатка от деления a на b)
17    return a % b >= 0 ? a % b : a % b + b; //проверяем, чтобы остаток был адекватным, если он отрицательный
18}
19
20int min(int a, int b) { //Функция поиска минимума из двух чисел
21    return a <= b ? a : b;
22}
23
24int max(int a, int b) { //Функция поиска максимума из двух чисел
25    return a >= b ? a : b;
26}
27
28int i_next(int i, int j, int k, int l) { //Функция изменения координаты i(x)
29    return abs(k - 15) - min(i / 3, mod((j + l), 10)) - 20;
30}
31
32int j_next(int i, int j, int k, int l) { //Функция изменения координаты j(y)
33    return -1 * (j + k) / 5 + abs(mod(i * l, 8));
34}
35
36int l_next(int i, int j, int k, int l) { //Функция изменения динамического параметра движения l
37    return max(mod((i + j), 15), mod((l + k), 14));
38}
39
40bool check(int i, int j) { //Функция проверки попадания точки в нужную область
41    return (((i >= CENTRE_I - (SQUARE_SIDE / 2)) && (i <= CENTRE_I + (SQUARE_SIDE / 2))) && ((j >= CENTRE_J - (SQUARE_SIDE / 2)) && (j <= CENTRE_J + (SQUARE_SIDE / 2))));
42}
43
44int main() { //Основная функция
45    int i = I0, j = J0, l = L0, k; //Присваиваем координатам и динамическому параметру движения начальные значения из констант
46    int counter_popadaniy = 0; //Вводим счётчик попаданий в нужную область
47    for (k = 0; k < K_MAX; k++) { //Запускаем цикл от 0 до 49 с шагом 1, итератор-k
48        i = i_next(i, j, k, l); //Обновляем координаты и параметр движения
49        j = j_next(i, j, k, l);
50        l = l_next(i, j, k, l);
51        if (check(i, j)) { //Проверяем попадание в нужную область
52            printf("Попадание в заданную область, i=%d, j=%d, k=%d, l=%d\n", i, j, k, l); //Выводим значения, при которых мы попали в нужную область
53            counter_popadaniy++; //Увеличиваем счётчик попаданий в нужную область
54        }
55    }
56    if (counter_popadaniy == 0) { //Если счётчик попаданий остался равен нулю, мы выводим значения на 50 (последнем) шаге
57        k++;
58        i = i_next(i, j, k, l);
59        j = j_next(i, j, k, l);
60        l = l_next(i, j, k, l);
61        printf("Количество шагов исчерпано, i=%d, j=%d, k=%d, l=%d\n", i, j, k, l);
62    }
63    return 0;
64}
```

```
alexandr@alexandr-Magicbook16:~$ gcc Lab.c
```

```
alexandr@alexandr-Magicbook16:~$ ./a.out
```

```
Попадание в заданную область, i=10, j=-6, k=45, l=4
```

```
Попадание в заданную область, i=8, j=-8, k=46, l=8
```

```
Попадание в заданную область, i=12, j=-7, k=47, l=13
```

```
Попадание в заданную область, i=11, j=-7, k=49, l=13
```

```
alexandr@alexandr-Magicbook16:~$
```

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 составлять программы с динамическими вычислениями на языке Си, научился работать с некоторыми библиотеками, константами, функциями, циклами и условными операторами - развил свои умения в программировании на языке Си.

Недочёты при выполнении задания могут быть устранены следующим образом: _____

Подпись студента