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

Студент группы М80-103Б-21 Березнев Никита Вадимович, № по списку 4

Контакты e-mail: nikita.berezneff@yandex.ru

Работа выполнена: «22» октября 2021г.

Преподаватель: каф. 805 Севастьянов Виктор Сергеевич

Отчет сдан « » \_\_\_\_\_\_\_\_\_20\_\_ г., итоговая оценка \_\_\_\_\_\_\_\_

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. **Тема:** Системы программирования на языке Си.

1. **Цель работы:** составление и отладка простейшей программы на языке Си итеративного характера с целочисленными рекуррентными соотношениями, задающими некоторое регулярное движение точки в целочисленной системе координат (i, j) с дискретным временем k и динамическим параметром движения l.
2. **Задание (вариант 4):** Вывести сообщение об итоге движения: попадание в заданную область плоскости не более чем за 50 шагов или сообщение о промахе. Также в результат надо включить время окончания движения, конечные координаты точки, значение динамического параметра движения и номер шага итерации.
3. **Оборудование** (студента):

Процессор *Intel Core i3-3217U @ 4x 1.80GH* с ОП *7851* Мб, НМД *512* Гб. Монитор *1366x768*

1. **Программное обеспечение (**студента**):**

Операционная система семейства: *linux*, наименование: *ubuntu*, версия *18.10 cosmic*

интерпретатор команд: *bash* версия *4.4.19*.

Система программирования -- версия --

Утилиты операционной системы:

Прикладные системы и программы --

Местонахождение и имена файлов программ и данных на домашнем компьютере --

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

- Поскольку формулы подсчета переменных представлены в условии лабораторной работы, то все, что необходимо для решения данной задачи – это реализовать представленный алгоритм. Однако для упрощения работы с условиями подсчета переменных стоит описать ряд функций, введение которых значительно облегчит восприятие кода и, как следствие, значительно сократит количество ошибок в коде программы.

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

- Интерпретировать рекуррентные соотношения, являющиеся формулами подсчёта переменных, из условия лабораторной работы на язык Си.

- Описать ряд функций для оптимизации тела основной программы.

- Опираясь на интерпретированное условие задачи, а также на функции, описанные ранее, создать программу.

- Составить ряд тестов, с помощью которых можно было бы проверить корректность работы составленного алгоритма.

|  |  |
| --- | --- |
| **Входные данные** | **Выходные данные** |
| 0 0 0 | No  0 0 0 50 |
| 1 1 2 | No  0 66 0 50 |
| 5 10 4 | Yes  16 15 10 2 |
| -9 -4 3 | Yes  17 13 6 1 |

**8. Распечатка протокола** (подклеить листинг окончательного варианта программы с тестовыми примерами,подписанныйпреподавателем).

#include <stdio.h>

#include <math.h>

int abs(int N)

{

if (N >= 0) {

return N;

} else {

return (N \* (-1));

}

}

int divf(double M, double x)

{

double N;

N = M / x;

if (x >= 0) {

return floor(N);

} else {

return ceil(N);

}

}

int mod(int N, int x)

{

return (N - divf(N, x) \* x);

}

int min(int I\_J\_old, int I\_L\_old)

{

if (I\_J\_old < I\_L\_old) {

return I\_J\_old;

} else {

return I\_L\_old;

}

}

int max(int I\_J\_old, int I\_L\_old, int J\_L\_old)

{

if ((I\_J\_old >= I\_L\_old) && (I\_J\_old >= J\_L\_old)) {

return I\_J\_old;

} if ((I\_L\_old >= I\_J\_old) && (I\_L\_old >= J\_L\_old)) {

return I\_L\_old;

} else {

return J\_L\_old;

}

}

int sign(int J\_old)

{

if (J\_old > 0) {

return 1;

} if (J\_old < 0) {

return -1;

} else {

return 0;

}

}

int main(void)

{

int I\_old;

int J\_old;

int L\_old;

int I\_new;

int J\_new;

int L\_new;

scanf("%d", &I\_new);

scanf("%d", &J\_new);

scanf("%d", &L\_new);

if ((((10 - I\_new) \* (10 - I\_new) + (10 - J\_new) \* (10 - J\_new)) <= 100) && (((10 - I\_new) \* (10 - I\_new) + (10 - J\_new) \* (10 - J\_new)) > 25)) {

printf("Yes\n%d %d %d %d\n", I\_new, J\_new, L\_new, 0);

} else {

for (int k = 1 ; k < 51 ; k++) {

I\_old = I\_new;

J\_old = J\_new;

L\_old = L\_new;

I\_new = mod((min(I\_old + J\_old, I\_old + L\_old) \* (k)), 30);

J\_new = J\_old + mod(L\_old \* sign(J\_old), 20) + mod((k - 1) \* sign(I\_old), 10);

L\_new = mod(max(I\_old \* J\_old, I\_old \* L\_old, J\_old \* L\_old), 30);

if ((((10 - I\_new) \* (10 - I\_new) + (10 - J\_new) \* (10 - J\_new)) <= 100) && (((10 - I\_new) \* (10 - I\_new) + (10 - J\_new) \* (10 - J\_new)) > 25)) {

printf("Yes\n%d %d %d %d\n", I\_new, J\_new, L\_new, k);

break;

}

if (k >= 50) {

printf("No\n%d %d %d %d\n", I\_new, J\_new, L\_new, k);

}

}

}

return 0;

}

**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 других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Лаб. | Дата | Время | Событие | Действие по исправлению | Примечание |
|  | или |  |  |  |  |  |
|  | дом. |  |  |  |  |  |
| 1 | дом | 29.10.21 | 2:46 | При прохождении чекера программа выпадала в ошибку “ Нарушение правил оформления программ ” | Изменение кода программы согласно всем канонам кодстайла | Борьба с чекером за пропуск программы каждый раз превращается в челлендж под названием “Борьба кожаного мешка с бездушной машиной” |
| 2 | дом | 29.10.21 | 18:21 | При прохождении чекера программа выпадала в ошибку “Неправильный ответ” | Вне зависимости от случая необходимо выводить 4 переменные | Вот и верь после такого строке “Формат результата” в чекере. Я всегда знал, что чекер – бездушная цифровая последовательность… |
| 3 | дом | 29.10.21 | 23:10 | При прохождении чекера программа выпадала в ошибку “Неправильный ответ” | В одной из формул нахождения переменной исправить I на J | Последние несколько часов программа не работала из-за того, что я перепутал переменную…  \*Вдох\* АААААААААААААААА |
| 4 | дом | 29.10.21 | 23:37 | При прохождении чекера программа выпадала в ошибку “Неправильный ответ” | Исключение границ внутренней окружности из множества точек кольца | \*Человек, проведший все лабы с чекером\*  “Этот бедняга понимает, что жизнь и страдание – одно и то же.”  @Предок |

1. **Замечания автора** по существу работы

Работу считаю весьма полезной, очень интересной и в то же время необычайно выматывающей (Спасибо, чекер, за то, что ты есть). Создавая программы на Си с элементами рекуррентных отношений, я обрел много новых знаний не только в синтаксисе языка, но и в самом механизме работы рекуррентных отношений. Ну и, конечно же, продолжил закреплять синтаксис языка с помощью чекера (Каждое полное прохождение чекера на новой лабе достойно получить прозвище “Путь боли”).

1. **Выводы**

Работа мне понравилась (особенно, когда чекер показал, что моя программа работает корректно). Было довольно занимательно 60+ раз переписывать алгоритм. Благодаря данной работе я освоил механизмы работы программы итеративного характера с рекуррентными соотношениями, что расширило область моих знаний. Информация, полученные мной в ходе выполнения данной лабораторной работы, поможет мне в дальнейшем при составлении более сложных алгоритмов.

Подпись студента \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_