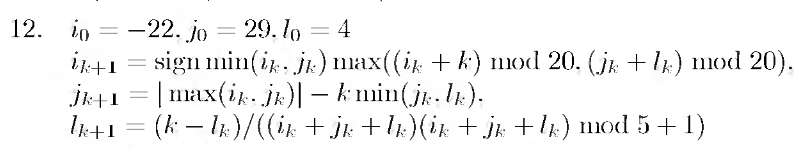
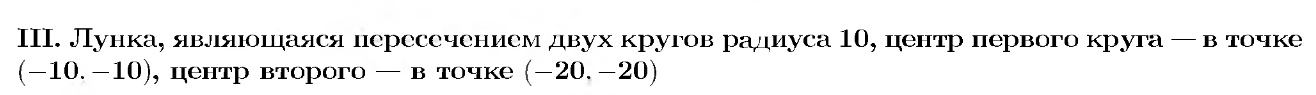
|  |  |
| --- | --- |
|  | **Отчёт по лабораторным работам** № 8, 9  по курсу: Фундаментальная информатика  студента группы : M8O-105Б-21 Козлова Егора Сергеевича , № по списку: 10  Контакты: e-mail iamaghoulzxc@gmail.com  Работа выполнена: “19“ ноября 2021 г.  Преподаватель: каф. 806 В. К. Титов  Входной контроль знаний с оценкой  Отчёт сдан: “19“ ноября 2021 г., итоговая оценка  Подпись преподавателя |

1. **Тема**: Cистемы программирования на языке Си. Программирование на языке Си.
2. **Цель работы**: изучение конкретной системы программирования на языке Си и получение навыков подготовки тестов и отладки программ; составление и отладка простейшей программы на языке Си итеративного характера с целочисленными рекуррентными соотношениями.
3. **Задание** (*вариант №* 12):

продемонстрировать работу компилятора языка Си на примере любой программы, написанной на Си. Покажите, какие сообщения выдаёт компилятор при синтаксических ошибках в программе, и что происходит при аварийных ошибках (например, при делении на ноль).



1. **Оборудование** (*лабораторное*):

*Не использовалось*

*Оборудование ПЭВМ студента, если использовалось:*

Процессор AMD Ryzen 7 5800H @ 8x 3.2GHz , ОП 15429 МБ, НМД 1024 ГБ. Монитор: встроенный (1920x1080)

1. *Программное обеспечение ЭВМ студента, если использовалось:*

Операционная система семейства Linux, наименование: Ubuntu версия 20.04.3 LTS x86\_64 \_\_

Интерпретатор команд: bash версия 5.0.17

Редактор текстов: Emacs версия 26.3

Утилиты операционной системы:

Прикладные системы и программы: gcc

Местонахождения и имена файлов программ и данных: /bin

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

ЛР 8:

1. Создание программы с ошибками

2. Попытаться скомпилировать программу с помощью компилятора gcc

3. Исправить ошибки

4.Попытаться скомпилировать программу с помощью компилятора gcc

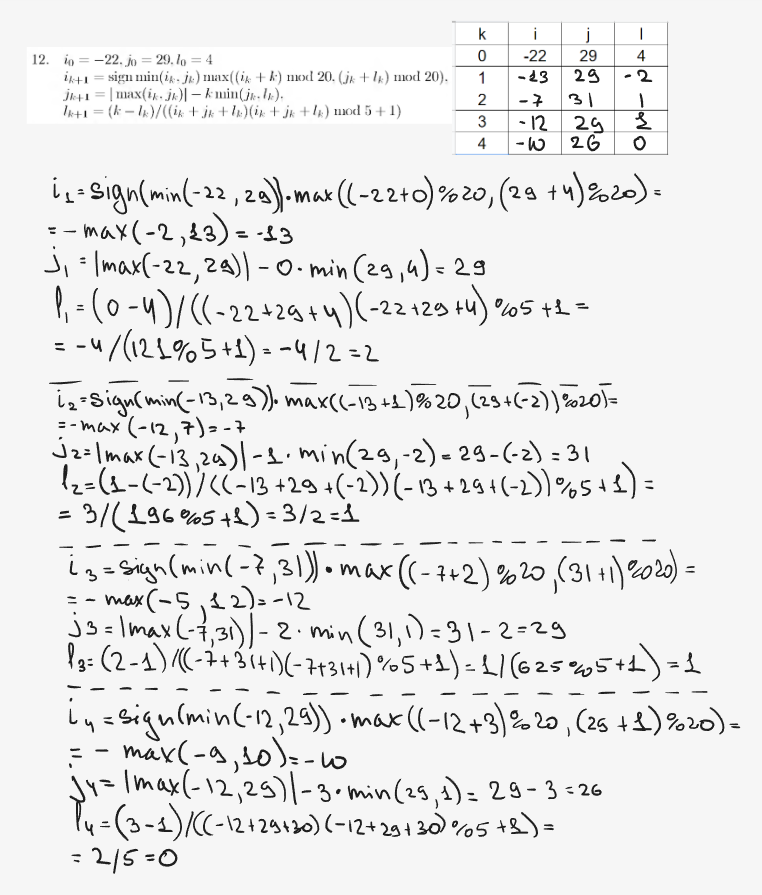
ЛР 9:

1. Выполнить ручной просчет данных рекуррентных соотношений.

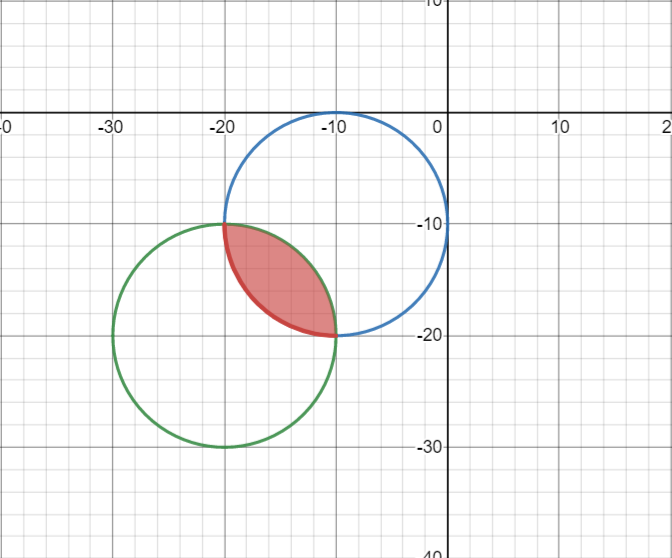
2. Написать программу и выполнить отладку в терминале

3. Сравнить результаты в терминале с ручным просчетом, если возникли ошибки, то нужно исправить их

4. Внести в таблицу первые 4 шага просчета.



Ручной просчет:



← область попадания(выделена красным)

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

Лабораторная работа №8:

#include <stdio.h>  
  
void func() {  
 return 5;  
}  
  
int func1() {  
 return "test";  
}  
  
int main() {  
 int i, n = 10;  
 for (i = 0; i < 10; i++) printf("%d\n", i / 0);  
 func();  
 func1();  
  
 return 0  
}

Лабораторная работа №9:

#include <stdio.h>  
  
int sign(int n) {  
 if (n > 0) return 1;  
 else if (n < 0) return -1;  
 else return n;  
}  
  
int max(int n, int m) {  
 if (n > m) return n;  
 else return m;  
}  
  
int min(int n, int m) {  
 if (n < m) return n;  
 else return m;  
}  
  
int abs(int n) {  
 if (n < 0) return -n;  
 else return n;  
}  
  
int in\_circle(int n, int m) { //функция, проверяющая попадание точки в область  
 if (((n + 10) \* (n + 10) + (m + 10) \* (m + 10) <= 100) && ((n + 20) \* (n + 20) + (m + 20) \* (m + 20) <= 100))  
 return 1;  
 else return 0;  
}  
  
  
int main() {  
 const int i0 = -22, j0 = 29, l0 = 4;  
 int i = i0, j = j0, l = l0, k, ii, jj, ll;  
 for (k = 0; k < 50; k++) {  
 if (in\_circle(i, j)) {  
 printf("point get\ntime k = %d\ncord i = %d\ncord j = %d\ndynamic l = %d\n", k, i, j,  
 l); //вывод если попала  
 break;  
 }  
 ii = i;  
 jj = j;  
 ll = l;  
 i = sign(min(ii, jj)) \* max((ii + k) % 20, (jj + ll) % 20);  
 j = abs(max(ii, jj)) - k \* min(jj, ll);  
 l = (k - ll) / ((ii + jj + ll) \* (ii + jj + ll) % 5 + 1);  
  
 }  
 if (k == 50)  
 printf("point doesn't get :(\ntime k = %d\ncord i = %d\ncord j = %d\ndynamic l = %d\n", k, i, j,  
 l); //вывод если не попала  
 return 0;  
}

Допущен к выполнению работы. Подпись преподавателя

1. **Распечатка протокола** (подклеить листинг окончательного варианта программы с текстовыми примерами, подписанный преподавателем)

**isitmuse@isitmuse**:**~/lab/ЛАБЫ/8-9**$ cat head8.txt

\_.\_ \_,-'""`-.\_

(,-.`.\_,'( |\`-/|

`-.-' \ )-`( , o o)

`- \`\_`"'-

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|||ЛАБОРАТОРНАЯ РАБОТА №8|||

||СИСТЕМЫ ПРОГРАММИРОВАНИЯ|||

|||||||||НА ЯЗЫКЕ СИ|||||||||

|||ВЫПОЛНИЛ СТУДЕНТ ГРУППЫ|||

|||М8О-105Б-21 КОЗЛОВ ЕГОР|||

\\\\\\\\\\\\\\|//////////////

**isitmuse@isitmuse**:**~/lab/ЛАБЫ/8-9**$ cat 8.c

#include <stdio.h>

void func() {

return 5;

}

int func1() {

return "test";

}

int main() {

int i, n = 10;

for (i = 0; i < 10; i++) printf("%d\n", i / 0);

func();

func1();

return 0

}

**isitmuse@isitmuse**:**~/lab/ЛАБЫ/8-9**$ gcc -o 8.out 8.c

**8.c:** In function ‘**func**’:

**8.c:4:12:** **warning:** ‘**return**’ with a value, in function returning void

4 | return **5**;

| **^**

**8.c:3:6:** **note:** declared here

3 | void **func**() {

| **^~~~**

**8.c:** In function ‘**func1**’:

**8.c:8:12:** **warning:** returning ‘**char \***’ from a function with return type ‘**int**’ makes integer from pointer without a cast [**-Wint-conversion**]

8 | return **"test"**;

| **^~~~~~**

**8.c:** In function ‘**main**’:

**8.c:13:47:** **warning:** division by zero [**-Wdiv-by-zero**]

13 | for (i = 0; i < 10; i++) printf("%d\n", i **/** 0);

| **^**

**8.c:17:13:** **error:** expected ‘**;**’ before ‘**}**’ token

17 | return 0

| **^**

| ;

18 | }

| ~

**isitmuse@isitmuse**:**~/lab/ЛАБЫ/8-9**$ cat head9.txt

\_.\_ \_,-'""`-.\_

(,-.`.\_,'( |\`-/|

`-.-' \ )-`( , o o)

`- \`\_`"'-

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|||ЛАБОРАТОРНАЯ РАБОТА №9|||

|||СОСТАВЛЕНИЕ ПРОСТЕЙШЕЙ|||

||||ПРОГРАММЫ НА ЯЗЫКЕ СИ||||

|||ВЫПОЛНИЛ СТУДЕНТ ГРУППЫ|||

|||М8О-105Б-21 КОЗЛОВ ЕГОР|||

\\\\\\\\\\\\\\|//////////////

**isitmuse@isitmuse**:**~/lab/ЛАБЫ/8-9**$ cat 9.c

#include <stdio.h>

int sign(int n) {

if (n > 0) return 1;

else if (n < 0) return -1;

else return n;

}

int max(int n, int m) {

if (n > m) return n;

else return m;

}

int min(int n, int m) {

if (n < m) return n;

else return m;

}

int abs(int n) {

if (n < 0) return -n;

else return n;

}

int in\_circle(int n, int m) { //функция, проверяющая попадание точки в область

if (((n + 10) \* (n + 10) + (m + 10) \* (m + 10) <= 100) && ((n + 20) \* (n + 20) + (m + 20) \* (m + 20) <= 100))

return 1;

else return 0;

}

int main() {

const int i0 = -22, j0 = 29, l0 = 4;

int i = i0, j = j0, l = l0, k, ii, jj, ll;

for (k = 0; k < 50; k++) {

if (in\_circle(i, j)) {

printf("point get\ntime k = %d\ncord i = %d\ncord j = %d\ndynamic l = %d\n", k, i, j,

l); //вывод если попала

break;

}

ii = i;

jj = j;

ll = l;

i = sign(min(ii, jj)) \* max((ii + k) % 20, (jj + ll) % 20);

j = abs(max(ii, jj)) - k \* min(jj, ll);

l = (k - ll) / ((ii + jj + ll) \* (ii + jj + ll) % 5 + 1);

}

if (k == 50)

printf("point doesn&apos;t get :(\ntime k = %d\ncord i = %d\ncord j = %d\ndynamic l = %d\n", k, i, j,

l); //вывод если не попала

return 0;

}

**isitmuse@isitmuse**:**~/lab/ЛАБЫ/8-9**$ gcc -o 9.out 9.c

**isitmuse@isitmuse**:**~/lab/ЛАБЫ/8-9**$ ./9.out

point get

time k = 9

cord i = -12

cord j = -15

dynamic l = 5

1. **Дневник отладки** должен содержать дату и время сеансов отладки, и основные ошибки (ошибки в сценарии и программе, не стандартные операции) 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.  или  дом. | Дата | Время | Событие | Действие по исправлению | Примечание |
| 1 | Дом | 19.11.2021 | 19:53 | Лабораторная работа 8  1. Попытка вернуть число(int) (return) в функции void  2. Попытка вернуть символ(char) в функции int  3. Деление на ноль  4. Пропущена точка с запятой | Лабораторная работа 8  1. Изменить тип функции  2. Изменить тип функции  3. Убрать деление на ноль  4. Поставить точку с запятой |  |

1. Замечание автора по существу работы: замечания отсутствуют
2. Выводы:

В ходе данных лабораторных работ я научился писать простейшие программы итеративного характера с целочисленными рекуррентными значениями на языке Си. Также я научился компилировать программы и производить их отладку с помощью компилятора gcc.

Подпись студента