



Отчет по лабораторной работе №8-9 по курсу Фундаментальная информатика

Студент группы М80-101Б-21 Филимонов Николай Николаевич, № по списку 23

Контакты www, e-mail:

Работа выполнена: «» 2021 г.

Преподаватель: каф. 806 Титов В.К.

Входной контроль знаний с оценкой

Отчет сдан « » 2021 г., итоговая оценка

Подпись преподавателя

1. Тема: Программирование на языке СИ

2. Цель работы: составить и отладить простейшую программу на СИ. Изучить возможности компилятора.

3. Задание (вариант: 23): Движение точки задано рекуррентными формулами. Выяснить, окажется ли за 50 итераций точка внутри заданной плоскости. Если окажется, вывести номер итерации, переменные, если нет, то вывести итоговые значения переменных.

Треугольник с вершинами (0, 10), (-10, 0), (-10, 20); начальные значения переменных: $i0 = -8$, $j0 = -5$, $l0 = 12$,

$$ik+1 = (ik^2 / (|jk-lk|+k+1) - jk^2 / (|ik-lk|+k+1)) \bmod 30$$

$$jk+1 = \text{sign } lk \min(ik, jk) - \text{sign } jk \max(ik, lk) + k$$

$$lk+1 = (ik - jk)(jk - lk)(lk - ik) \bmod 20$$

4. Оборудование(лабораторное):

ЭВМ, процессор, имя узла сети, с ОП, ГБ,

НМД, ГБ, терминал-адрес, принтер,

Другие устройства

Оборудование ПЭВМ студента, если использовалось:

Процессор AMD Ryzen 7 2700, с ОП 16 ГБ

Другие устройства

5. Программное обеспечение:

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 GNU/Linux, наименование Pop!_OS версия 21.04

интерпретатор команд GNOME Terminal версия 3.38.2

Система программирования

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

Ручной просчёт:

$i_0 = -8$
 $j_0 = -5$
 $l_0 = 12$
 $i_1 = ((-8)^2/(|-5-12|+1+1) - (-5)^2/(|-8-12|+1+1)) \bmod 30 = 2$
 $j_1 = \text{sign } 12 \min(-8, -5) - \text{sign } -5 \max(-8, 12) + 1 = 4$
 $l_1 = (-8 - (-5))(-5 - 12)(12 - (-8)) \bmod 20 = 0$
 $i_2 = (2^2/(|4-0|+2+1) - 4^2/(|2-0|+2+1)) \bmod 30 = -4$
 $j_2 = \text{sign } 0 \min(2, 4) - \text{sign } 4 \max(2, 0) + 2 = -1$
 $l_2 = (2 - 4)(4 - 0)(0 - 2) \bmod 20 = 16$
 $i_3 = ((-4)^2/(|-1-16|+3+1) - (-1)^2/(|-4-16|+3+1)) \bmod 30 = 0$
 $j_3 = \text{sign } 16 \min(-4, -1) - \text{sign } -1 \max(-4, 16) + 3 = 14$
 $l_3 = (-4 - (-1))(-1 - 16)(16 - (-4)) \bmod 20 = 0$
 $i_4 = (0^2/(|14-0|+4+1) - 14^2/(|0-0|+4+1)) \bmod 30 = 4$
 $j_4 = \text{sign } 0 \min(0, 14) - \text{sign } 14 \max(0, 0) + 4 = -19$
 $l_4 = (0 - 14)(14 - 0)(0 - 0) \bmod 20 = 3$
 $i_5 = (4^2/(|-19-3|+5+1) - (-19)^2/(|4-3|+5+1)) \bmod 30 = 15$
 $j_5 = \text{sign } 3 \min(4, -19) - \text{sign } -19 \max(4, 3) + 5 = 4$
 $l_5 = (4 - (-19))(-19 - 3)(3 - 4) \bmod 20 = 14$

k	i	j	l
0	-8	-5	12
1	2	4	0
2	-4	-1	16
3	0	14	0
4	4	-19	3
5	15	4	14

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

```
#include <stdio.h>
```

```
int sign(int x){
if(x>0) return 1;
else if(x==0) return 0;
else return -1;
}
int abs(int x){
return x*sign(x);
}
int min(int x1,int y1){
```

```

if(x1>y1) return y1;
else return x1;
}
int max(int x1, int y1){
if(x1>=y1) return x1;
else return y1;
}
int ifunc(int k, int i, int j, int l){
k--;
return (i*i/(abs(j-l)+k+1)-j*j/(abs(i-l)+k+1))%30;
}
int jfunc(int k, int i, int j, int l){
k--;
return sign(l)*min(i, j)-sign(j)*max(i, l) + k;
}
int lfunc(int k, int i, int j, int l){
return (i-j)*(j-l)*(l-i)%20;
}
int main() {
int i, j, l, k = 1, cou=0, i0=-8, j0 = -5, l0=12;
while(k<50){
i = ifunc(k, i0, j0, l0);
j = jfunc(k, i0, j0, l0);
l = lfunc(k, i0, j0, l0);
if(i<=0 && i>=0-10 && j>=i+10 && j<=-i+10){
printf("Попадание в заданую область\n");
printf("k=%d i=%d j=%d l=%d\n", k, i, j, l);
return 0;
}
i0 = i;
j0 = j;
l0 = l;
k++;
}
printf("Нет попадания в заданую область\n");
printf("k=%d i=%d j=%d l=%d\n", k, i, j, l);
return 0;
}

```

Допущен к выполнению работы. Подпись преподавателя _____

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

```
nikolay@SABAKA-LINUX:~$ cat> laba8-9.c
```

```
#include <stdio.h>
```

```
int sign(int x){
```

```
if(x>0) return 1;
```

```
else if(x==0) return 0;
```

```

else return -1;
}
int abs(int x){
return x*sign(x);
}
int min(int x1,int y1){
if(x1>y1) return y1;
else return x1;
}
int max(int x1, int y1){
if(x1>=y1) return x1;
else return y1;
}
int ifunc(int k, int i, int j, int l){
k--;
return (i*i/(abs(j-l)+k+1)-j*j/(abs(i-l)+k+1))%30;
}
int jfunc(int k, int i, int j, int l){
k--;
return sign(l)*min(i, j)-sign(j)*i, l + k;
}
int lfunc(int k, int i, int j, int l){
return (i-j)*(j-l)*(l-i)%20;
}
int main() {
int i, j, l, k = 1, cou=0, i0=-8, j0 = -5, l0=12;
while(k<50){
i = ifunc(k, i0, j0, l0);
j = jfunc(k, i0, j0, l0);
l = lfunc(k, i0, j0, l0);
if(i<=0 && i>=0-10 && j>=i+10 && j<=-i+10){
printf("Попадание в заданую область\n");
printf("k=%d i=%d j=%d l=%d\n", k, i, j, l);
return 0;
}
i0 = i;
j0 = j;
l0 = l;
k++;
}
printf("Нет попадания в заданую область\n");
printf("k=%d i=%d j=%d l=%d\n", k, i, j, l);
return 0;
}
nikolay@SABAKA-LINUX:~$ g++ laba8-9.c
nikolay@SABAKA-LINUX:~$ ./a.out
Попадание в заданую область
k=13 i=-3 j=12 l=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. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание

10. Замечания автора :

11. Выводы

__ В ходе лабораторной работе я научился составлять и отлаживать простейшие программы на ЯП СИ, изучил возможности компилятора _____

Недочёты при выполнении задания могут быть устранены следующим образом:

Подпись студента _Постнов_____