



Отчёт по лабораторной работе № 9 по курсу 1

студента группы М80-108Б-19 Хренниковой Ангелины, № по списку 23

Адреса www, e-mail, jabber, skype: lina.khrennikova@mail.ru

Работа выполнена: "14" ноября 2019г.

Преподаватель: Поповкин А. В. каф.806

Входной контроль знаний с оценкой

Отчёт сдан "20" ноября 2019 г., итоговая оценка

Подпись преподавателя

1. Тема: Системы программирования на языке Си.

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

3. Задание (вариант №23): Вывести сообщение об итоге движения.

4. Оборудование (лабораторное):
ЭВМ PC, процессор Intel® Core™ i7-3770 CPU @ 3.40GHz * 8, имя узла сети alise18 с ОП 15974,4 МБ, НМД 345,5 ГБ.
Терминал Gnome адрес 192.168.2.118/24. Принтер
Другие устройства

Оборудование ПЭВМ студента, если использовалось:

Процессор Intel® Core™ i3-7020U CPU @ 2.30GHz * 4, ОП 8192 МБ, НМД 256 ГБ. Монитор LCD
Другие устройства

5. Программное обеспечение (лабораторное):
Операционная система семейства UNIX, наименование Ubuntu версия 18.04
Интерпретатор команд Bash версия 4.4.20(1)
Система программирования версия
Редактор текстов Nano версия 2.9.3
Утилиты операционной системы
Прикладные системы и программы
Местонахождения и имена файлов программ и данных

Программное обеспечение ЭВМ студента, если использовалось:

Операционная система семейства UNIX, наименование Ubuntu версия 18.04
Интерпретатор команд Bash версия 4.4.19(1)
Система программирования версия
Редактор текстов Emacs версия 25.2.2

Утилиты операционной системы	
Прикладные системы и программы	
Местонахождения и имена файлов программ и данных	

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

Для каждого движения считаем координату точки и динамический параметр движения (проходим в массиве от 1 до 50, считая для каждого шага i, j, l по предложенной формуле, реализовав функции \max и \min через if) и проверяем на нахождение в необходимой области через уравнения прямых.

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

Составила и отладила простейшую программу на Си в терминальном классе. Код программы:

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

int main()
{
    int i[51], j[51], l[51];
    i[0]=29, j[0]=-6, l[0]=1;
    for (int t=1; t<51; ++t) {
        if (i[t-1]-j[t-1]<i[t-1]-l[t-1]) {
            if (i[t-1]-j[t-1]>j[t-1]-l[t-1]) {
                if (i[t-1]-j[t-1]<i[t-1]-t) {
                    i[t]=(i[t-1]-j[t-1])%30;
                }
                else i[t]=(i[t-1]-t)%30;
            }
            else {
                if (j[t-1]-l[t-1]<i[t-1]-t) {
                    i[t]=(j[t-1]-l[t-1])%30;
                }
                else i[t]=(i[t-1]-t)%30;
            }
        }
        else {
            if (i[t-1]-l[t-1]>i[t-1]-l[t-1]) {
                if (i[t-1]-l[t-1]<i[t-1]-t) {
                    i[t]=(i[t-1]-l[t-1])%30;
                }
                else i[t]=(i[t-1]-t)%30;
            }
            else {
                if (i[t-1]-l[t-1]<i[t-1]-t) {
                    i[t]=(i[t-1]-l[t-1])%30;
                }
                else i[t]=(i[t-1]-t)%30;
            }
        }
    }

    for (int t=1; t<51; ++t) {
        if (i[t-1]-j[t-1]>i[t-1]-l[t-1]) {
            if (i[t-1]-j[t-1]<j[t-1]-l[t-1]) {
                if (i[t-1]-j[t-1]>i[t-1]-t) {
                    j[t]=(i[t-1]-j[t-1])%30;
                }
                else j[t]=(i[t-1]-t)%30;
            }
            else {
                if (i[t-1]-l[t-1]<i[t-1]-t) {
                    i[t]=(i[t-1]-l[t-1])%30;
                }
                else i[t]=(i[t-1]-t)%30;
            }
        }
        else {
            if (i[t-1]-l[t-1]>i[t-1]-l[t-1]) {
                if (i[t-1]-l[t-1]<i[t-1]-t) {
                    i[t]=(i[t-1]-l[t-1])%30;
                }
                else i[t]=(i[t-1]-t)%30;
            }
            else {
                if (i[t-1]-l[t-1]<i[t-1]-t) {
                    i[t]=(i[t-1]-l[t-1])%30;
                }
                else i[t]=(i[t-1]-t)%30;
            }
        }
    }
}
```

```

        if (j[t-1]-l[t-1]>i[t-1]-t) {
            j[t]=(j[t-1]-l[t-1])%30;
        }
        else j[t]=(i[t-1]-t)%30;
    }
}
else {
    if (i[t-1]-l[t-1]<i[t-1]-l[t-1]) {
        if (i[t-1]-l[t-1]>i[t-1]-t) {
            j[t]=(i[t-1]-l[t-1])%30;
        }
        else j[t]=(i[t-1]-t)%30;
    }
    else {
        if (i[t-1]-l[t-1]>i[t-1]-t) {
            j[t]=(i[t-1]-l[t-1])%30;
        }
        else j[t]=(i[t-1]-t)%30;
    }
}
}
}

l[t]=i[t-1]%30-j[t-1]%30+l[t-1]%30-t%30;
if (-10<=-i[t]-j[t] && 10<=j[t]-i[t] && i[t]>=-10) {
    printf("YES\n%d\n%d %d %d",t,i[t],j[t],l[t]); break;}
else if (t==50) {
    printf("NO\n50\n%d %d %d",i[50],j[50],l[50]); break;}
}
return 0;
}

```

Через функцию mod:

```

#include <stdio.h>
#include <math.h>

int mod(int a, int b) {
    int c=a-floor(a/b)*b;
}

int main()
{
    int i[51], j[51], l[51];
    i[0]=29, j[0]=-6, l[0]=1;
    for (int t=1; t<51; ++t) {
        if (i[t-1]-j[t-1]<i[t-1]-l[t-1]) {
            if (i[t-1]-j[t-1]>j[t-1]-l[t-1]) {
                if (i[t-1]-j[t-1]<i[t-1]-t) {
                    i[t]=mod(i[t-1]-j[t-1], 30);
                }
                else i[t]=mod(i[t-1]-t, 30);
            }
        }
        else {
            if (j[t-1]-l[t-1]<i[t-1]-t) {
                i[t]=mod(j[t-1]-l[t-1], 30);
            }
            else i[t]=mod(i[t-1]-t, 30);
        }
    }
}
else {
    if (i[t-1]-l[t-1]>i[t-1]-l[t-1]) {
        if (i[t-1]-l[t-1]<i[t-1]-t) {
            i[t]=mod(i[t-1]-l[t-1], 30);
        }
    }
}
}

```

```

        else i[t]=mod(i[t-1]-t, 30);
    }
    else {
        if (i[t-1]-l[t-1]<i[t-1]-t) {
            i[t]=mod(i[t-1]-l[t-1], 30);
        }
        else i[t]=mod(i[t-1]-t, 30);
    }
}

for (int t=1; t<51; ++t) {
    if (i[t-1]-j[t-1]>i[t-1]-l[t-1]) {
        if (i[t-1]-j[t-1]<j[t-1]-l[t-1]) {
            if (i[t-1]-j[t-1]>i[t-1]-t) {
                j[t]=mod(i[t-1]-j[t-1], 30);
            }
            else j[t]=mod(i[t-1]-t, 30);
        }
        else {
            if (j[t-1]-l[t-1]>i[t-1]-t) {
                j[t]=mod(j[t-1]-l[t-1], 30);
            }
            else j[t]=mod(i[t-1]-t, 30);
        }
    }
    else {
        if (i[t-1]-l[t-1]<i[t-1]-l[t-1]) {
            if (i[t-1]-l[t-1]>i[t-1]-t) {
                j[t]=mod(i[t-1]-l[t-1], 30);
            }
            else j[t]=mod(i[t-1]-t, 30);
        }
        else {
            if (i[t-1]-l[t-1]<i[t-1]-l[t-1]) {
                if (i[t-1]-l[t-1]>i[t-1]-t) {
                    j[t]=mod(i[t-1]-l[t-1], 30);
                }
                else j[t]=mod(i[t-1]-t, 30);
            }
            else {
                if (i[t-1]-l[t-1]>i[t-1]-t) {
                    j[t]=mod(i[t-1]-l[t-1], 30);
                }
                else j[t]=mod(i[t-1]-t, 30);
            }
        }
    }
}

//int c=mod(5, -3);

l[t]=i[t-1]%30-j[t-1]%30+l[t-1]%30-t%30;
if (-10<=-i[t]-j[t] && 10<=j[t]-i[t] && i[t]>=-10) {
    printf("YES\nk=%d\ni=%d j=%d l=%d\n",t,i[t],j[t],l[t]); break;}
else if (t==50) {
    printf("NO\nk=50\ni=%d j=%d l=%d\n",i[50],j[50],l[50]); break;}
}
return 0;
}

```

Пункты 1-7 отчёта составляются **строго до** начала лабораторной работы.

Допущен к выполнению работы. Подпись преподавателя _____

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

```
lina_tucha@LAPTOP-44CRFC1U:~$ ssh -p 6789 198201@172.16.80.213
198201@172.16.80.213's password:
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.15.0-65-generic x86_64)
```

```
* Documentation: https://help.ubuntu.com
* Management:   https://landscape.canonical.com
* Support:      https://ubuntu.com/advantage
```

Могут быть обновлены 125 пакетов.
0 обновлений касаются безопасности системы.

New release '18.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

```
*** Требуется перезагрузка системы ***
Last login: Tue Nov 12 10:08:22 2019 from 172.31.44.26
198201@cameron:~$ touch 2.c
198201@cameron:~$ nano 2.c
198201@cameron:~$ gcc 2.c -o superhwoolen
198201@cameron:~$ ./superhwoolen
YES
7
-2 12 -39
198201@cameron:~$ nano 2.c
198201@cameron:~$ gcc 2.c -o superhwoolen
198201@cameron:~$ ./superhwoolen
YES
7
-2 12 -39
39198201@cameron:~$ nano 2.c
198201@cameron:~$ gcc 2.c -o superhwoolen
198201@cameron:~$ ./superhwoolen
YES
k=7
i=-2 j=12 l=-39
```

9. **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

10. Замечание автора по существу работы _____

11. Выводы : Составила и отладила простейшую программу на языке Си итеративного характера с целочисленными рекуррентными соотношениями, задающими некоторое регулятивное движение точки в целочисленной системе координат (i,j) с дискретным временем k и динамическим параметром движения l.

Недочеты, допущенные при выполнении задания, могут быть устранены следующим образом _____

Подпись студента Хренникова А. С.