Лабораторна робота №5. Циклічні конструкції

Індивідульне завдання на оцінку “добре” номер 2.

Реалізувати програму відповідно до індивідуального завдання за допомогою трьох типів циклів : *for, while_do, do_while* (отримати три однакових результата)

1. Вимоги

1) Розробник

- Князькін Владислав Ігорович
- студент групи КІТ-320
  09.11.20

2) Загальне завдання

Визначити зворотне число для заданого цілого числа. Кількість розрядів від самого початку не визначено.

2. Хід роботи

1. Зробимо у раніше створеному репозиторії нову під-директорію *lab04.*
   valadon@valadon-VirtualBox:~ PROGRAMMING-KNIAZKIN/naxxramass $ mkdir lab05
2. Створимо папку для нашого завдання *lab05_2* та відкриємо у запустимо у ньому *nano.*
   valadon@valadon-VirtualBox:~ PROGRAMMING-KNIAZKIN/naxxramass $ mkdir lab05_2
   valadon@valadon-VirtualBox:~ PROGRAMMING-KNIAZKIN/naxxramass/lab05/lab05_2 $ nano
3. Записуємо наш код програми, дотримуючись всіх відступів та закриття дужок.
   1. Запишемо нашу програму:

```
//*      Цикл For


void main() {  //*создаем функцию
   int n = 123456;      //* создаем переменную изначального числа
   int m;         //* создаем переменную перевернутого числа
   int n1 , n2 , n3 , n4 , n5, n6;         //* создаем переменные цифр числа
   n1 = n / 100000;    //* формула для расчета первого числа
   n2 = n / 10000 % 10;        //* формула для расчета второго числа
   n3 = n / 1000 % 10;         //* формула для расчета третьего числа
   n4 = n / 100 % 10;  //* формула для расчета четвертого числа
   n5 = n / 10 % 10;   //* формула для расчета пятого числа
   n6 = n % 10;         //* формула для расчета шестого числа
   for ( int ten = 10; n6 != (n % 10) * 100000 ; ten == 10){ //* создаем цикл для расчета первой цифры перевернутого числа,пока n6 не равно (n % 10)*100000
   n6 = n6 * ten;        //* приравниваем n6 к выше указанному значению, чтобы завершить цикл
   }     //* завершение цикла
```

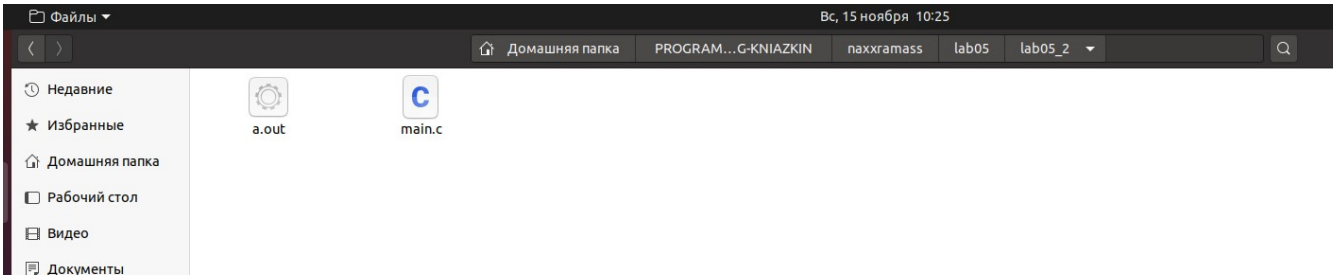
```
   if ( n6 != (n % 10) * 100000 ){      //* создаем условие , если n6 не равно (n % 10) * 100000
         for ( int ten1 = 10; n6 != n % 10;){    //*создаем цикл для корректировки первой цифры перевернутого числа
         n6 = n6 / ten1;//* корректируем первую цифру
         }         //* цикл завершен
         }else{  //* иначе
         m = n6 + m;    //* вписываем первую цифру в перевернутое число
         }         //* условие завершено
   for ( int ten = 10; n5 != (n / 10 % 10) * 10000; ten == 10){        //* создаем цикл для расчета второй цифры перевернутого числа
   n5 = n5 * ten;         //* приравниваем n5 к выше указанному значению, чтобы завершить цикл
   }      //* завершение цикла
   if ( n5 != (n / 10 % 10) * 10000 ){//* создаем условие , если n5 не равно (n / 10 % 10) * 10000
         for ( int ten1 = 10; n5 != n / 10 % 10;){        //*создаем цикл для корректировки второй цифры перевернутого числа
         n5 = n5 / ten1;//* корректируем вторую цифру, если n5 != (n / 10 % 10) * 10000
         }         //* цикл завершен
         }else{  //*иначе
         m = n5 + m;    //* вписываем вторую цифру в перевернутое число
         }         //* условие завершено
   for ( int ten = 10; n4 != (n / 100 % 10) * 1000; ten == 10){        //* создаем цикл для расчета третьей цифры перевернутого числа
   n4 = n4 * ten;         //* приравниваем n4 к выше указанному значению, чтобы завершить цикл
   }      //* завершение цикла
   if ( n4 != (n / 100 % 10) * 1000 ){//* создаем условие , если n4 не равно (n / 100 % 10) * 1000
         for ( int ten1 = 10; n4 != n / 100 % 10;){      //*создаем цикл для корректировки третьей цифры перевернутого числа
         n4 = n4 / ten1;//* корректируем третью цифру, если n4 != (n / 100 % 10) * 1000
         }         //* цикл завершен
         }else{  //*иначе
         m = n4 + m;    //* вписываем третью цифру в перевернутое число
         }         //* условие завершено
   for ( int ten = 10; n3 != (n / 1000 % 10) * 100; ten == 10){        //* создаем цикл для расчета следующей цифры перевернутого числа
   n3 = n3 * ten;         //* приравниваем n3 к выше указанному значению, чтобы завершить цикл
   }      //* завершение цикла
   if ( n3 != (n / 1000 % 10) * 100 ){//* создаем условие , если n3 не равно (n / 1000 % 10) * 100
         for ( int ten1 = 10; n3 != n / 100 % 10;){      //*создаем цикл для корректировки следующей цифры перевернутого числа
         n3 = n3 / ten1;//* корректируем следующую цифру, если n3 != (n / 1000 % 10) * 100
         }         //* цикл завершен
         }else{  //*иначе
         m = n3 + m;    //* вписываем следующую цифру в перевернутое число
         }         //* условие завершено
   for ( int ten = 10; n2 != (n / 10000 % 10) * 10; ten == 10){        //* создаем цикл для расчета следующей цифры перевернутого числа
   n2 = n2 * ten;         //* приравниваем n2 к выше указанному значению, чтобы завершить цикл
   }      //* завершение цикла
   if ( n2 != (n / 10000 % 10) * 10 ){//* создаем условие , если n2 не равно (n / 10000 % 10) * 10
```

```
        for ( int ten1 = 10; n2 != n / 10000 % 10;){  //*создаем цикл для корректировки следующей цифры перевернутого числа
        n2 = n2 / ten1;//* корректируем следующую цифру, если n2 != (n / 10000 % 10) * 10
        }       //* цикл завершен
        }else{  //*иначе
        m = n2 + m;    //* вписываем следующую цифру в перевернутое число
        }       //* условие завершено
    for ( int ten = 10; n1 != n / 100000; ten == 10){   //* создаем цикл для расчета последней цифры перевернутого числа
    n1 = n1 * ten;      //* приравниваем n1 к выше указанному значению, чтобы завершить цикл
    }   //* завершение цикла
        if ( n1 != n / 100000 ){        //* создаем условие , если n1 не равно n / 100000
                for (int ten1 = 10; n1 != n / 100000;){         //*создаем цикл для корректировки последней цифры перевернутого числа
                n1 = n1 / ten1;//* корректируем последнюю цифру, если n1 != n / 100000
                }       //* цикл завершен
                }else { //*иначе
                n1 = n / 100000;        //* вписываем последнюю цифру в перевернутое число
                }       //* условие завершено
    m = n6 + n5 + n4 + n3 + n2 + n1; //* объединяем цифры в обратное число
        //*завершаем циел For



//* Цикл while_do

    int m_w;    //* создаем переменную перевернутого числа
    int n1_w, n2_w , n3_w , n4_w , n5_w, n6_w;      //* создаем переменные цифр числа
    n1_w = n / 100000;//* формула для расчета первого числа
    n2_w = n / 10000 % 10;    //* формула для расчета второго числа
    n3_w = n / 1000 % 10;     //* формула для расчета третьего числа
    n4_w = n / 100 % 10;      //* формула для расчета четвертого числа
    n5_w = n / 10 % 10;       //* формула для расчета пятого числа
    n6_w = n % 10;    //* формула для расчета шестого числа
        while ( m_w != n6_w + n5_w + n4_w + n3_w + n2_w + n1_w ){  //* создаем цикл while с условием m != n6 + n5 + n4 + n3 + n2 + n1
                while ( n6_w != (n % 10) * 100000 ){        //* создаем цикл while с условием n6 != (n % 10) * 100000
                n6_w = n6_w * 10;    //* приравниваем n6 к выше указанному значению, чтобы завершить цикл
                }       //* завершаем цикл
                while ( n5_w != (n / 10 % 10) * 10000 ){    //* создаем цикл while с условием n5 != (n / 10 % 10) * 10000
                n5_w = n5_w * 10;    //* приравниваем n5 к выше указанному значению, чтобы завершить цикл
                }       //* завершаем цикл
                while ( n4_w != (n / 100 % 10) * 1000 ){    //* создаем цикл while с условием n4 != (n / 100 % 10) * 1000
```

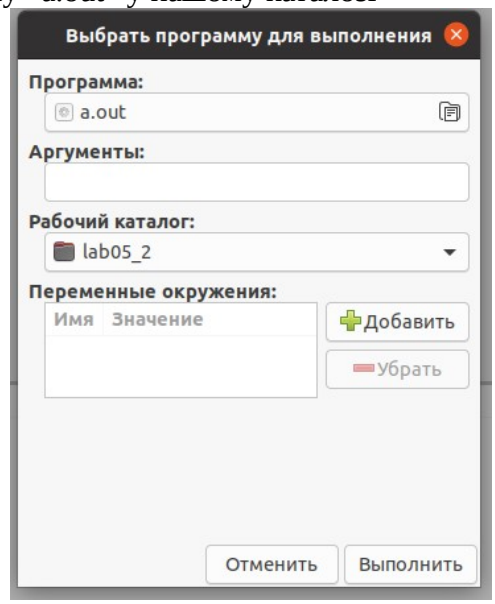
```
                n4_w = n4_w * 10;    //* приравниваем n4 к выше указанному значению, чтобы
завершить цикл
                }        //* завершаем цикл
                while ( n3_w != (n / 1000 % 10) * 100 ){     //* создаем цикл while с условием n3 != (n
/ 1000 % 10) * 100
                n3_w = n3_w * 10;    //* приравниваем n3 к выше указанному значению, чтобы
завершить цикл
                }        //* завершаем цикл
                while ( n2_w != (n / 10000 % 10) * 10 ){     //* создаем цикл while с условием n2 != (n
/ 10000 % 10) * 10
                n2_w = n2_w * 10;    //* приравниваем n2 к выше указанному значению, чтобы
завершить цикл
                }        //* завершаем цикл
                while ( n1_w != n / 100000){//* создаем цикл while с условием n1 != n / 100000
                n1_w = n1_w * 10;    //* приравниваем n1 к выше указанному значению, чтобы
завершить цикл
                        if ( n1_w != n / 100000 ){      //* создаем условие , если n1 не равно n / 100000
                        while ( n1_w != n / 100000){//* создаем цикл while с условием n1 != n /
100000
                        n1_w = n1_w / 10;     //* корректируем последнюю цифру, если n1 != n /
100000
                        }        //* завершаем цикл
                        }else { //*иначе
                        n1_w = n / 100000;    //* вписываем последнюю цифру в перевернутое число
                        }        //* завершаем условие
                }        //* завершаем цикл
        m_w = n6_w + n5_w + n4_w + n3_w + n2_w + n1_w;       //* объединяем цифры в обратное
число
        }        //* завершение цикла
        //* завершаем цикл While_do


//* Цикл do_while




int m_d;        //* создаем переменную перевернутого числа
   int n1_d , n2_d , n3_d , n4_d , n5_d, n6_d;          //* создаем переменные цифр числа
   n1_d = n / 100000; //* формула для расчета первого числа
   n2_d = n / 10000 % 10;     //* формула для расчета второго числа
   n3_d = n / 1000 % 10;      //* формула для расчета третьего числа
   n4_d = n / 100 % 10;       //* формула для расчета четвертого числа
   n5_d = n / 10 % 10;        //* формула для расчета пятого числа
   n6_d = n % 10;      //* формула для расчета шестого числа
        do {    //* создаем цикл do
                do {    //* создаем цикл do
                n6_d = n6_d * 10;      //* приравниваем n6 к выше указанному значению, чтобы
завершить цикл
```

```
		}while ( n6_d != (n % 10) * 100000 );		//* условие цикла do ,n6_d != (n % 10) * 100000
		do {	//* создаем цикл do
		n5_d = n5_d * 10;	//* приравниваем n5 к выше указанному значению, чтобы завершить цикл
		}while ( n5_d != (n / 10 % 10) * 10000 );	//* условие цикла do ,n5_d != (n / 10 % 10) * 10000
		do {	//* создаем цикл do
		n4_d = n4_d * 10;	//* приравниваем n4 к выше указанному значению, чтобы завершить цикл
		}while ( n4_d != (n / 100 % 10) * 1000 );	//* условие цикла do ,n4_d != (n / 100 % 10) * 1000
		do {	//* создаем цикл do
		n3_d = n3_d * 10;	//* приравниваем n3 к выше указанному значению, чтобы завершить цикл
		}while ( n3_d != (n / 1000 % 10) * 100 );	//* условие цикла do ,n3_d != (n / 1000 % 10) * 100
		do {	//* создаем цикл do
		n2_d = n2_d * 10;	//* приравниваем n2 к выше указанному значению, чтобы завершить цикл
		}while ( n2_d != (n / 10000 % 10) * 10 );	//* условие цикла do ,n2_d != (n / 10000 % 10) * 10
		do {	//* создаем цикл do
		n1_d = n1_d * 10;	//* приравниваем n1 к выше указанному значению, чтобы завершить цикл
			if ( n1_d != n / 100000 ){	//* создаем условие , если n1_d не равно n / 100000
			do {	//* создаем цикл do
			n1_d = n1_d / 10;	//* корректируем последнюю цифру, если n1 != n / 100000
			}while ( n1_d != n / 100000);		//* условие цикла do ,n1_d != n / 100000
			}else{	//* иначе
			n1_d = n / 100000;	//* вписываем последнюю цифру в перевернутое число
			}		//* завершаем условие
		}while ( n1_d != n / 100000);		//* условие цикла do , n1_d != n / 100000
		m_d = n6_d + n5_d + n4_d + n3_d + n2_d + n1_d; //* объединяем цифры в обратное число
	}while ( m_d != n6_d + n5_d + n4_d + n3_d + n2_d + n1_d );	//* условие цикла do , m_d != n6_d + n5_d + n4_d + n3_d + n2_d + n1_d

}	//* завершаем цикл Do_while
```

4. Збережемо наш код та підпишемо його *main.c.*
5. Повертаємось до терміналу. Завдяки команді (у каталозі lab05):
   gcc -g main.c //* Зробимо виконувальні файли для наших *main.c*
   Вони автоматично назвуться a.out



6. Запускаємо відлагодник (nemiver), щоб перевірити, як працює наша програма. Щоб запустити програму, треба запустити її через виконувальний файл:
   1. Відкриваємо “Файл”
   2. Знаходимо фунцію “Завантажити виконувальний файл...”
   3. Вибираємо програму “a.out” у нашому каталозі



   4. Натискаємо “Виконати”.
   Тепер можна побачити програму, яку ми написали раніше:
   Завдяки функції “Step over”, переводимо нашу “стрілку” до *останнього рядка* ,щоб перевірити результат команди.

main.c

```c
//*     Цикл For

void main() {   //*создаем функцию
    int n = 123456;     //* создаем переменную изначального числа
    int m;      //* создаем переменную перевернутого числа
    int n1 , n2 , n3 , n4 , n5, n6;     //* создаем переменные цифр числа
    n1 = n / 100000;   //* формула для расчета первого числа
    n2 = n / 10000 % 10;        //* формула для расчета второго числа
    n3 = n / 1000 % 10;         //* формула для расчета третьего числа
    n4 = n / 100 % 10; //* формула для расчета четвертого числа
    n5 = n / 10 % 10;  //* формула для расчета пятого числа
    n6 = n % 10;        //* формула для расчета шестого числа
    for ( int ten = 10; n6 != (n % 10) * 100000 ; ten == 10){       //* создаем цикл для расчета первой цифры перевернутого числа,пока n6 не равно (n % 10)*100000
    n6 = n6 * ten;     //* приравниваем n6 к выше указанному значению, чтобы завершить цикл
    }   //* завершение цикла
    if ( n6 != (n % 10) * 100000 ){    //* создаем условие , если n6 не равно (n % 10) * 100000
        for ( int ten1 = 10; n6 != n % 10;){   //*создаем цикл для корректировки первой цифры перевернутого числа
        n6 = n6 / ten1;         //* корректируем первую цифру
        }       //* цикл завершен
        }else{  //* иначе
        m = n6 + m;     //* вписываем первую цифру в перевернутое число
        }       //* условие завершено
    for ( int ten = 10; n5 != (n / 10 % 10) * 10000; ten == 10){     //* создаем цикл для расчета второй цифры перевернутого числа
```

Строка: 83, Сто

ID по	Кадр	Функция	Аргументы	Распол
1	0	main	()	main.c

Переменная	Значение	Тип
Локальные переменные		
n	123456	int
m	654321	int
n1	1	int
n2	20	int
n3	300	int
n4	4000	int
n5	50000	int
n6	600000	int
m_w	0	int
n1_w	0	int
n2_w	0	int
n3_w	1431650368	int
n4_w	21845	int

main.c

```c

//* Цикл while_do

    int m_w;    //* создаем переменную перевернутого числа
    int n1_w, n2_w , n3_w , n4_w , n5_w, n6_w;      //* создаем переменные цифр числа
    n1_w = n / 100000; //* формула для расчета первого числа
    n2_w = n / 10000 % 10;      //* формула для расчета второго числа
    n3_w = n / 1000 % 10;       //* формула для расчета третьего числа
    n4_w = n / 100 % 10;        //* формула для расчета четвертого числа
    n5_w = n / 10 % 10;         //* формула для расчета пятого числа
    n6_w = n % 10;      //* формула для расчета шестого числа
        while ( m_w != n6_w + n5_w + n4_w + n3_w + n2_w + n1_w ){     //* создаем цикл while с условием m != n6 + n5 + n4 + n3 + n2 + n1
                while ( n6_w != (n % 10) * 100000 ){  //* создаем цикл while с условием n6 != (n % 10) * 100000
                n6_w = n6_w * 10;       //* приравниваем n6 к выше указанному значению, чтобы завершить цикл
                }       //* завершаем цикл
                while ( n5_w != (n / 10 % 10) * 10000 ){       //* создаем цикл while с условием n5 != (n / 10 % 10) * 10000
                n5_w = n5_w * 10;       //* приравниваем n5 к выше указанному значению, чтобы завершить цикл
                }       //* завершаем цикл
                while ( n4_w != (n / 100 % 10) * 1000 ){       //* создаем цикл while с условием n4 != (n / 100 % 10) * 1000
                n4_w = n4_w * 10;       //* приравниваем n4 к выше указанному значению, чтобы завершить цикл
                }       //* завершаем цикл
                while ( n3_w != (n / 1000 % 10) * 100 ){       //* создаем цикл while с условием n3 != (n / 1000 % 10) * 100
                n3_w = n3_w * 10;       //* приравниваем n3 к выше указанному значению, чтобы завершить цикл
```

Строка: 83, Столбец: 6

ID по	Кадр	Функция	Аргументы	Распол
1	0	main	()	main.c

Переменная	Значение	Тип
Локальные переменные		
n	123456	int
m	654321	int
n1	1	int
n2	20	int
n3	300	int
n4	4000	int
n5	50000	int
n6	600000	int
m_w	654321	int
n1_w	1	int
n2_w	20	int
n3_w	300	int
n4_w	4000	int

main.c <Disassembly>

```c
//* Цикл do_while


int m_d;        //* создаем переменную перевернутого числа
    int n1_d , n2_d , n3_d , n4_d , n5_d, n6_d;       //* создаем переменные цифр числа
    n1_d = n / 100000; //* формула для расчета первого числа
    n2_d = n / 10000 % 10;      //* формула для расчета второго числа
    n3_d = n / 1000 % 10;       //* формула для расчета третьего числа
    n4_d = n / 100 % 10;        //* формула для расчета четвертого числа
    n5_d = n / 10 % 10;         //* формула для расчета пятого числа
    n6_d = n % 10;      //* формула для расчета шестого числа
        do {    //* создаем цикл do
                do {    //* создаем цикл do
                n6_d = n6_d * 10;       //* приравниваем n6 к выше указанному значению, чтобы завершить цикл
                }while ( n6_d != (n % 10) * 100000 ); //* условие цикла do ,n6_d != (n % 10) * 100000
                do {    //* создаем цикл do
                n5_d = n5_d * 10;       //* приравниваем n5 к выше указанному значению, чтобы завершить цикл
                }while ( n5_d != (n / 10 % 10) * 10000 );      //* условие цикла do ,n5_d != (n / 10 % 10) * 10000
                do {    //* создаем цикл do
                n4_d = n4_d * 10;       //* приравниваем n4 к выше указанному значению, чтобы завершить цикл
```

Строка: 160, Столбец:

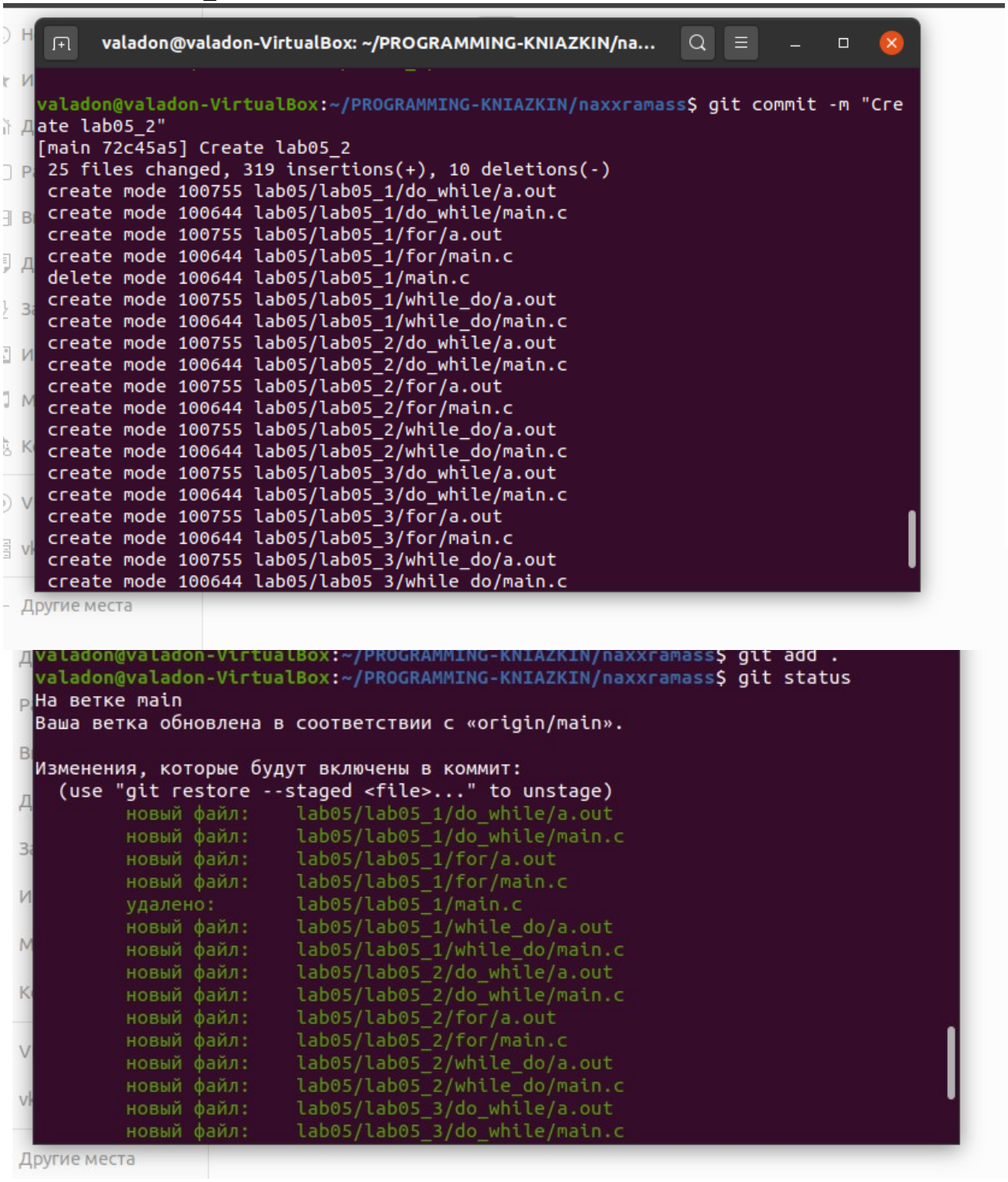
ID по	Кадр	Функция	Аргументы
1	0	main	()

Переменная	Значение	Тип
n5	50000	int
n6	600000	int
m_w	654321	int
n1_w	1	int
n2_w	20	int
n3_w	300	int
n4_w	4000	int
n5_w	50000	int
n6_w	600000	int
m_d	654321	int
n1_d	1	int
n2_d	20	int
n3_d	300	int

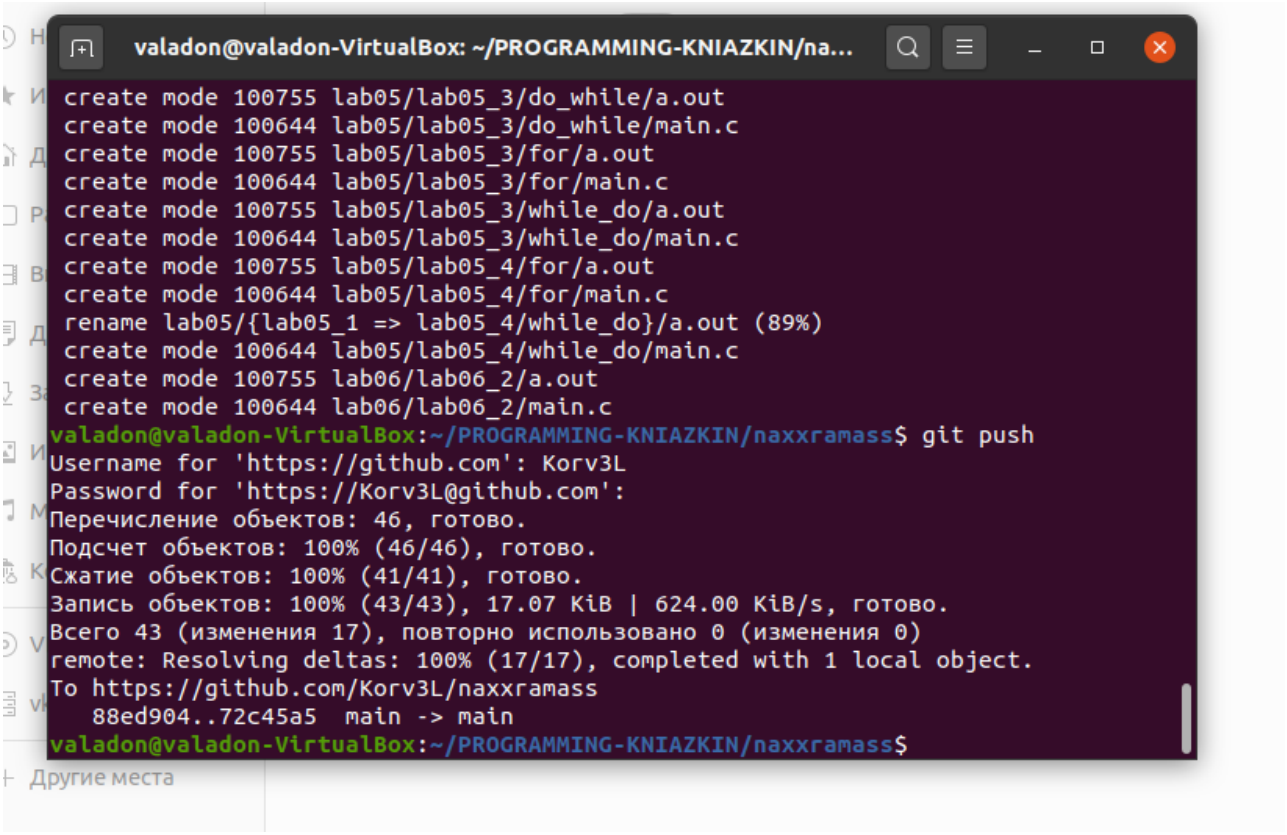
7. Бачимо, що результат вірний. Збережемо зміни у нашій директорії на *github* через команди *git*:

valadon@valadon-VirtualBox:~ PROGRAMMING-KNIAZKIN/naxxramass $ git add .
valadon@valadon-VirtualBox:~ PROGRAMMING-KNIAZKIN/naxxramass $ git status

valadon@valadon-VirtualBox:~ PROGRAMMING-KNIAZKIN/naxxramass $ git commit -m "Create lab05_2"



valadon@valadon-VirtualBox:~ PROGRAMMING-KNIAZKIN/naxxramass $ git push



3. **Способи використання**
   Дана програма являє собою зворотний порядок фільтру. Можна змінити порядок категорій. Наприклад: 1) Ліки — Мед.аппарати — Самозахист — Гігієна — Побутова хімія; 2) (у зворотньому) Побутова хімія — Гігієна — Самозахист — Мед.аппарати — Ліки
4. **Висновок**
   Завдяки цій лабораторній роботі, я навчився робити звичайні циклічні конструкції на язику C різними видами циклів(for, while_do, do_while), у системі Linux, додавати до них коментарі, перевіряти програму на дієздатність у відлагоднику (nemiver), та відправляти зміни у свій репозиторій на github.