МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**ОТЧЁТ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНОЙ РАБОТЫ № 1  
по дисциплине "Функциональное и логическое программирование"**

**Вариант 12**

Работу выполнил\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.В.Чухиль

Направление подготовки 02.03.02 Фундаментальная информатика и

информационные технологии

Направленность (профиль) Математическое и программное обеспечение  
 компьютерных технологий

Преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.А.Климец

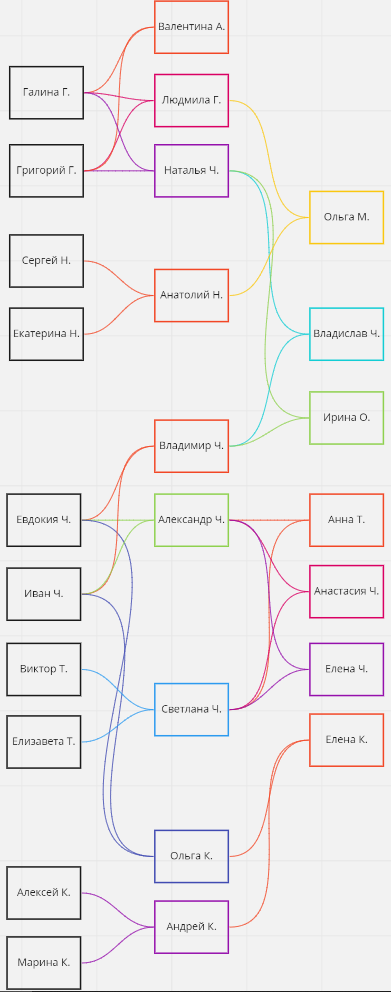
Краснодар

2020

**Задание 1.**

Построить дерево семьи в любом графическом редакторе, скопировать в отчет. Составить базу данных Prolog (набор фактов) о поле всех членов семьи (к пример, man, woman). Написать несколько запросов к терминалу на проверку пола конкретных членов семьи. Снимки экрана запросов прикрепить в отчет. Построить предикаты men и women, которые выводят на экран всех мужчин и всех женщин соответственно. Провести трассировку, снимок экрана прикрепить в отчет.

**Дерево семьи:**



**Код:**

man(grigory\_g).

man(ivan\_c).

man(anatoly\_n).

man(vladimir\_c).

man(alexander\_c).

man(andrei\_k).

man(vladislav\_c).

man(sergei\_n).

man(victor\_t).

man(alexei\_k).

woman(galina\_g).

woman(eudokia\_c).

woman(valentina\_a).

woman(ludmila\_g).

woman(natalia\_c).

woman(svetlana\_c).

woman(olga\_k).

woman(olga\_m).

woman(irina\_o).

woman(anna\_t).

woman(anastasia\_c).

woman(elena\_c).

woman(elena\_k).

woman(ekaterina\_n).

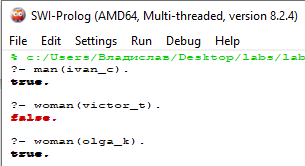
woman(elizaveta\_t).

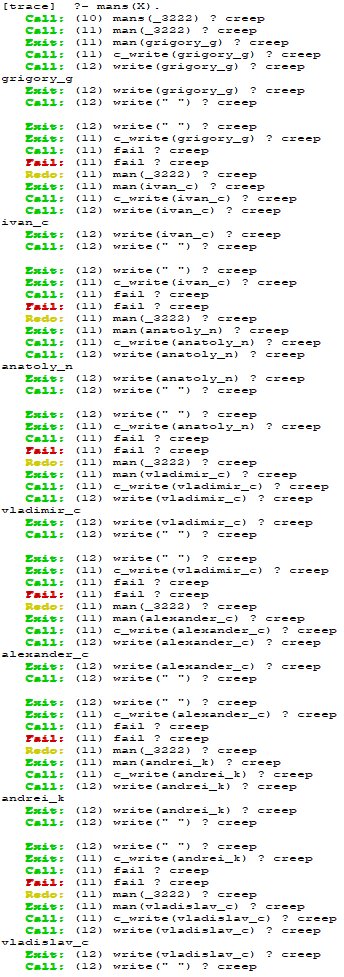
woman(marina\_k).

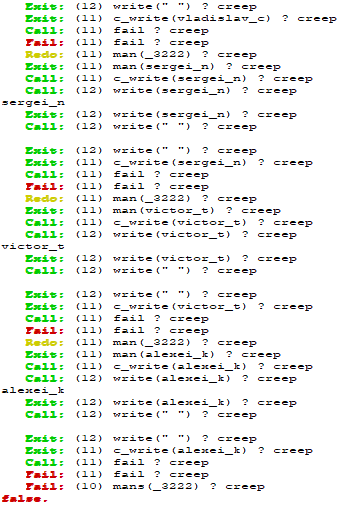
mans(X):-man(X),c\_write(X),fail.

womans(X):-woman(X),c\_write(X),fail.

**Скрины:**







**Задание 2.**

Построить базу данных Prolog отношения «является родителем», пример доступен по ссылке выше. Провести несколько запросов к терминалу Prolog, снимки экрана трех результатов прикрепить в отчет. Построить предикат, который children(X), который выводит всех детей X.

**Код:**

parent(galina\_g, valentina\_a).

parent(grigory\_g, valentina\_a).

parent(galina\_g, ludmila\_g).

parent(grigory\_g, ludmila\_g).

parent(galina\_g, natalia\_c).

parent(grigory\_g, natalia\_c).

parent(sergei\_n, anatoly\_n).

parent(ekaterina\_n, anatoly\_n).

parent(eudokia\_c, vladimir\_c).

parent(ivan\_c, vladimir\_c).

parent(eudokia\_c, alexander\_c).

parent(ivan\_c, alexander\_c).

parent(eudokia\_c, olga\_k).

parent(ivan\_c, olga\_k).

parent(victor\_t, svetlana\_c).

parent(elizaveta\_t, svetlana\_c).

parent(alexei\_k, andrei\_k).

parent(marina\_k, andrei\_k).

parent(anatoly\_n, olga\_m).

parent(ludmila\_g, olga\_m).

parent(natalia\_c, vladislav\_c).

parent(vladimir\_c, vladislav\_c).

parent(natalia\_c, irina\_o).

parent(vladimir\_c, irina\_o).

parent(alexander\_c, anna\_t).

parent(svetlana\_c, anna\_t).

parent(alexander\_c, anastasia\_c).

parent(svetlana\_c, anastasia\_c).

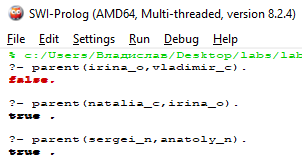
parent(alexander\_c, elena\_c).

parent(svetlana\_c, elena\_c).

parent(olga\_k, elena\_k).

parent(andrei\_k, elena\_k).

**Скрины:**



**Задание 5.**

Построить предикат brother(X, Y), который проверяет, является ли X братом Y. Построить предикат brothers(X), который выводит всех братьев X. Провести трассировку последнего предиката, добавить снимок экрана в отчет для трех разных запросов к терминалу.

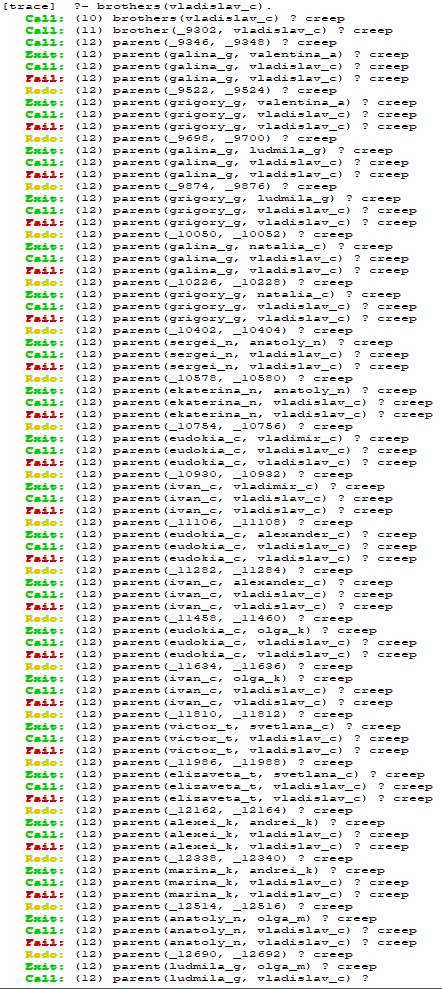
**Код:**

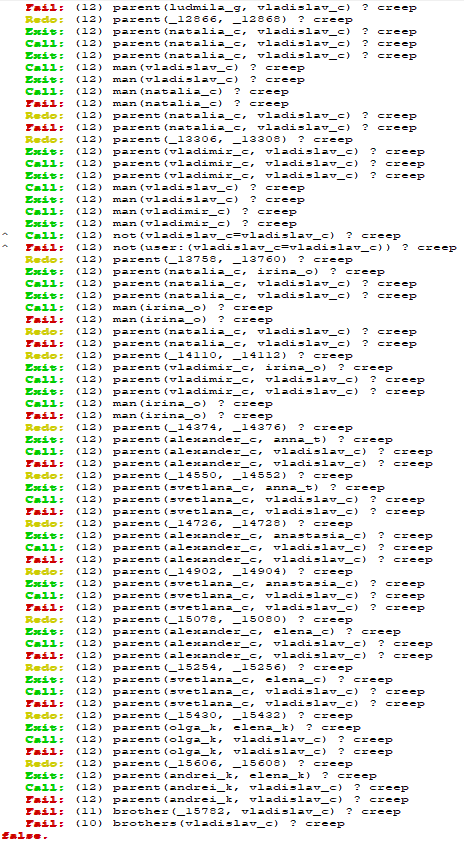
brother(X,Y):-parent(Z,X),parent(Z,Y),man(X),man(Z),not(X=Y).

brothers(X):-brother(Y,X),c\_write(Y),fail.

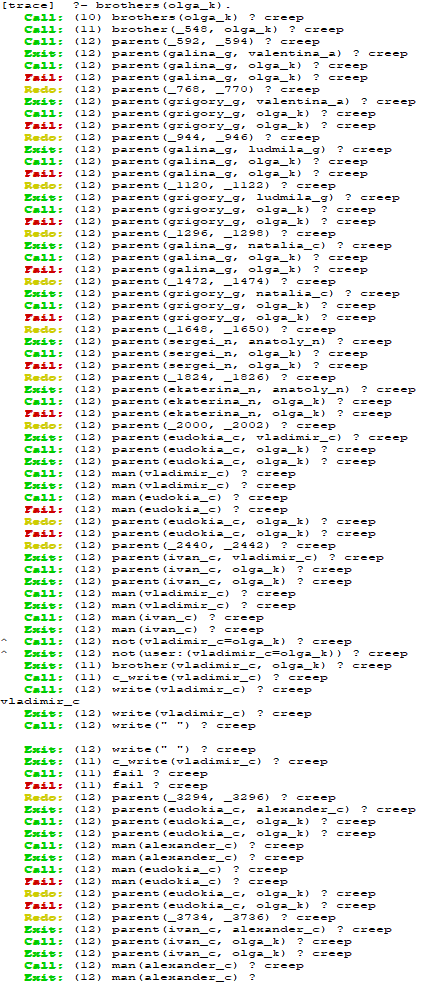
**Скрины:**

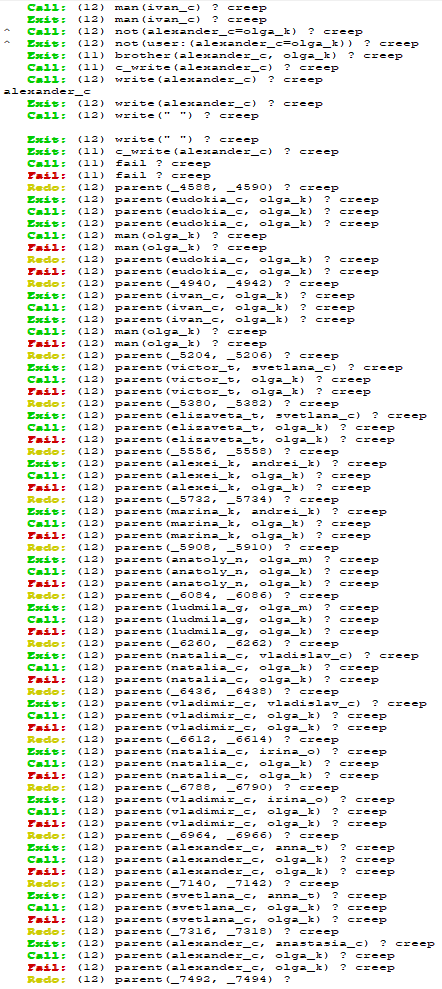
Пример 1.

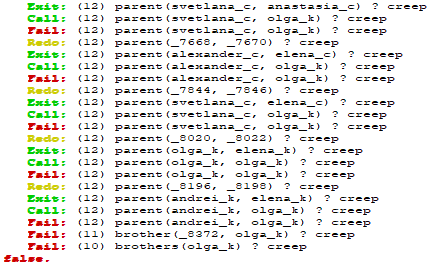




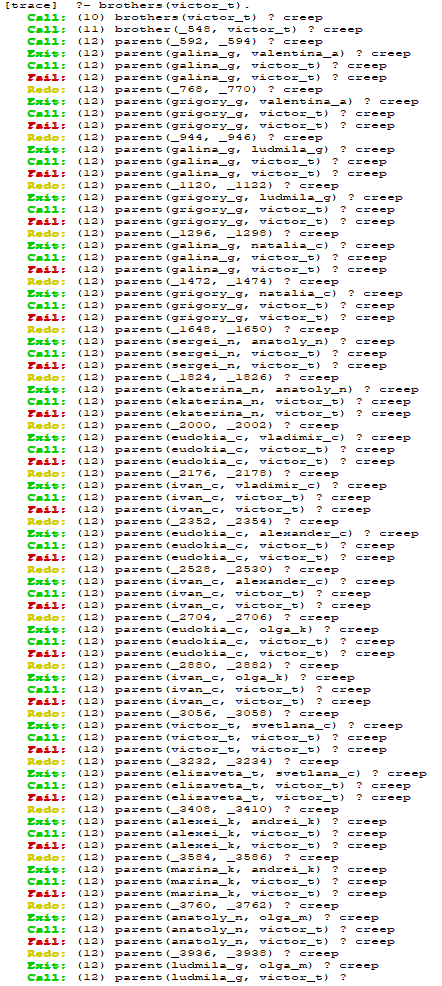
Пример 2.

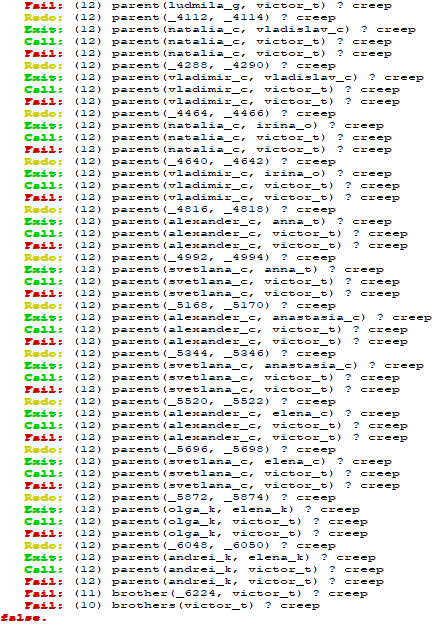






Пример 3.





**Задание 7.**

Построить предикат b\_s(X,Y), который проверяет, являются ли X и Y родными братом и сестрой или братьями или сестрами. Построить предикат b\_s(X), который выводит всех братьев или сестер X. Провести трассировку последнего предиката, добавить снимок экрана в отчет для трех разных запросов к терминалу.

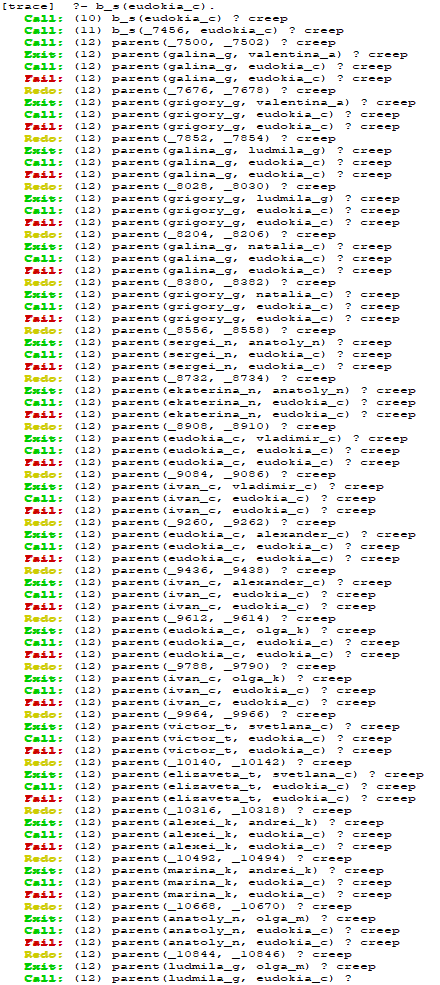
**Код:**

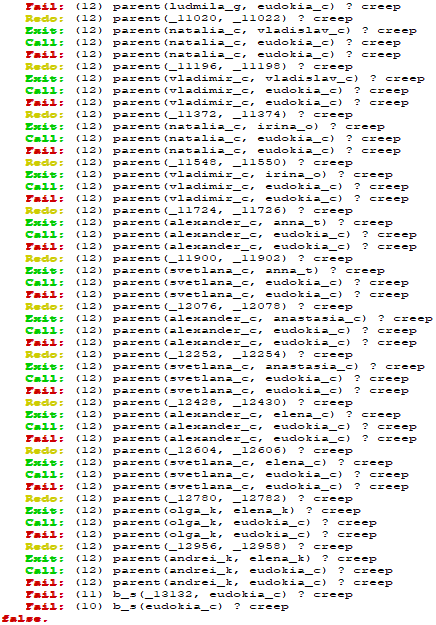
b\_s(X,Y):-parent(Z,X),parent(Z,Y),man(Z),not(X=Y).

b\_s(X):-b\_s(Y,X),c\_write(Y),fail.

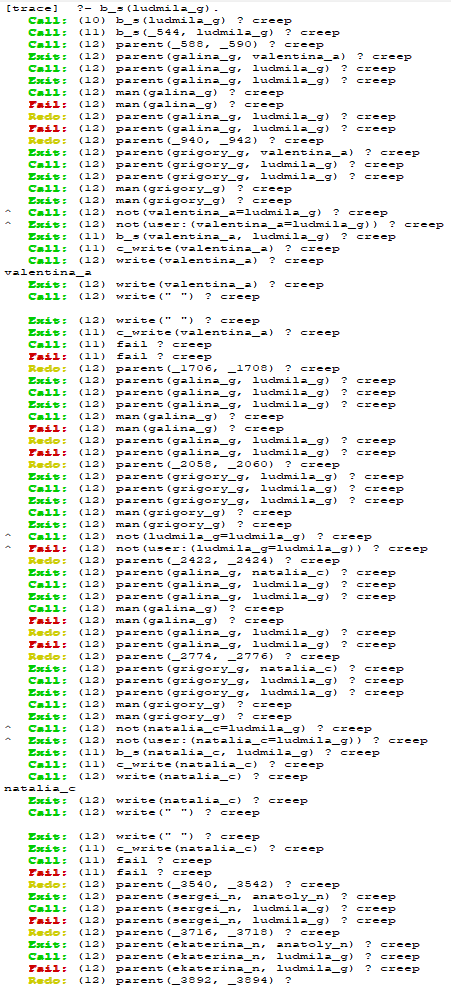
**Скрины:**

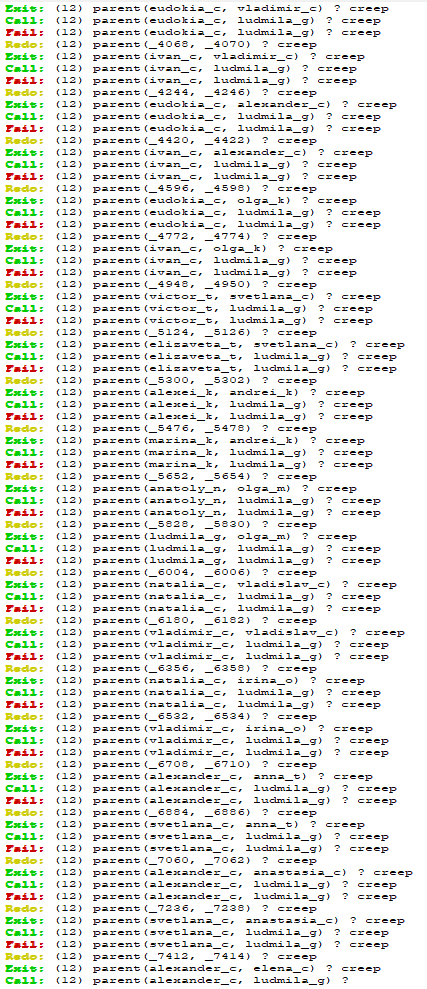
Пример 1.

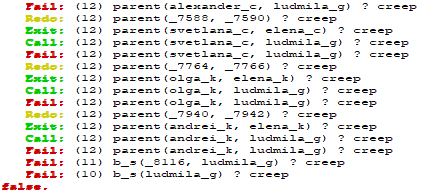




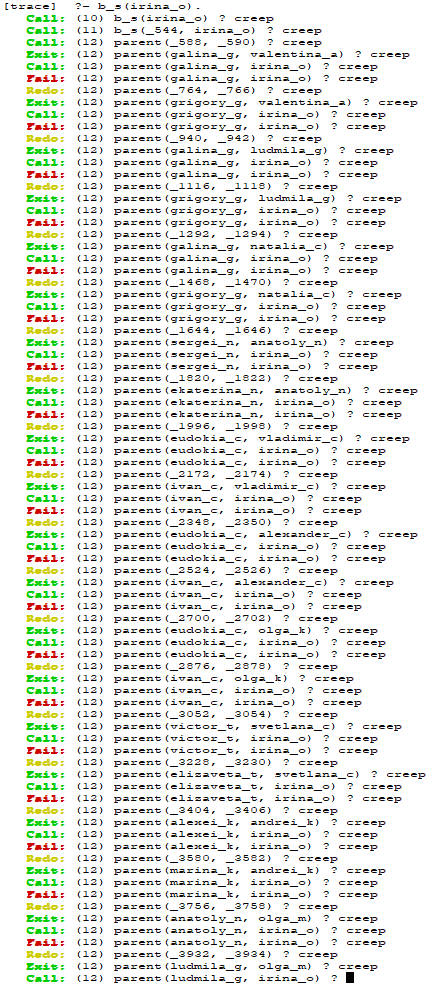
Пример 2.

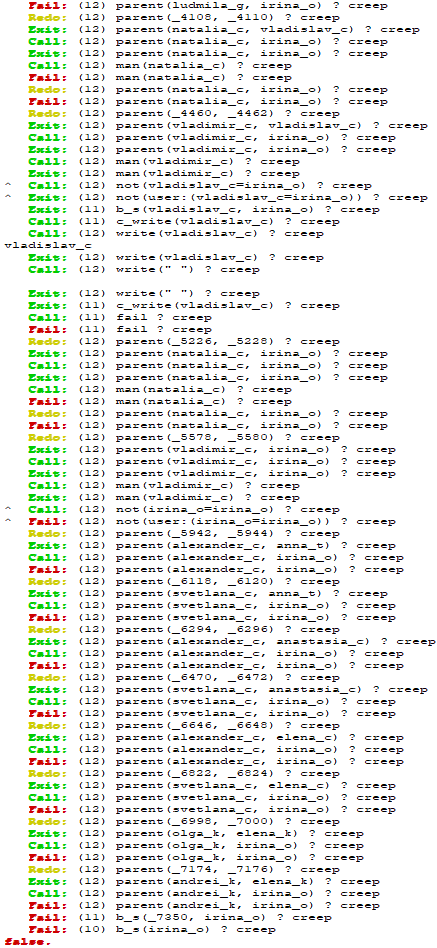






Пример 3.

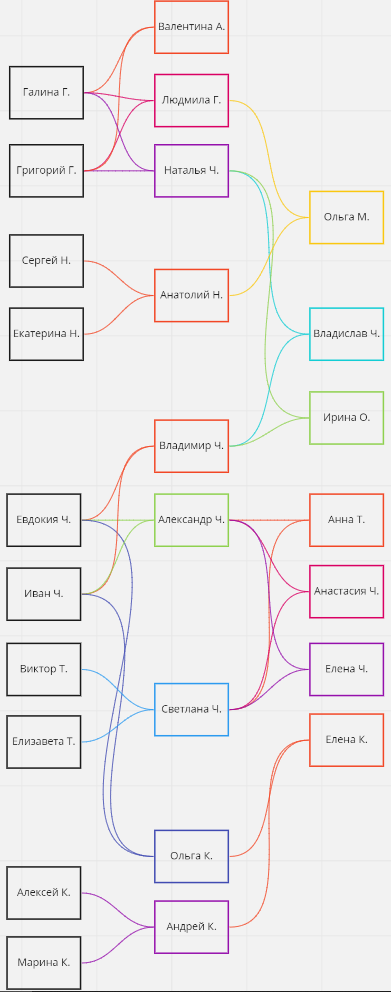




**Задание 8.**

Построить предикат grand\_pa(X, Y), который проверяет, является ли X дедушкой Y. Дополнить базу фактов таким образом, чтобы учитывать, что у каждого ребенка 2 бабушки и 2 дедушки. Внести изменения в дерево семьи, добавить дерево в отчет. Построить предикат grand\_pas(X), который выводит всех дедушек X. Провести трассировку последнего предиката, добавить снимок экрана в отчет для трех разных запросов к терминалу.

**Дерево семьи:**



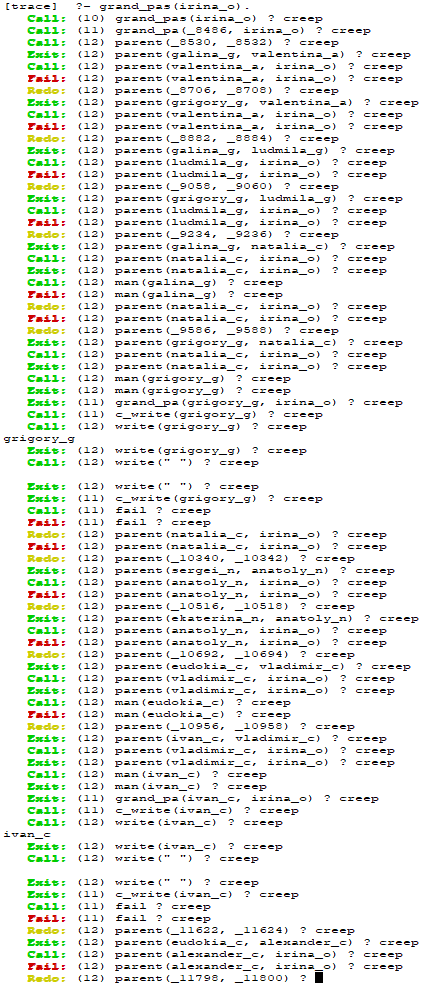
**Код:**

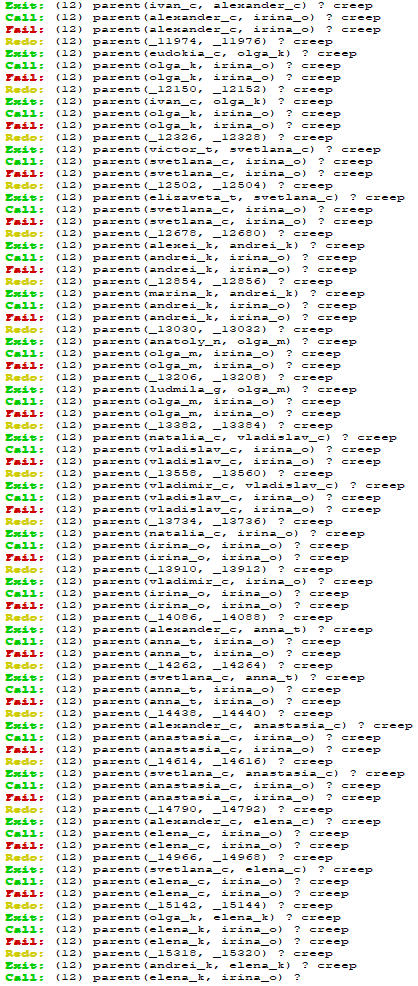
grand\_pa(X,Y):-parent(X,Z),parent(Z,Y),man(X).

grand\_pas(X):-grand\_pa(Y,X),c\_write(Y),fail.

**Скрины:**

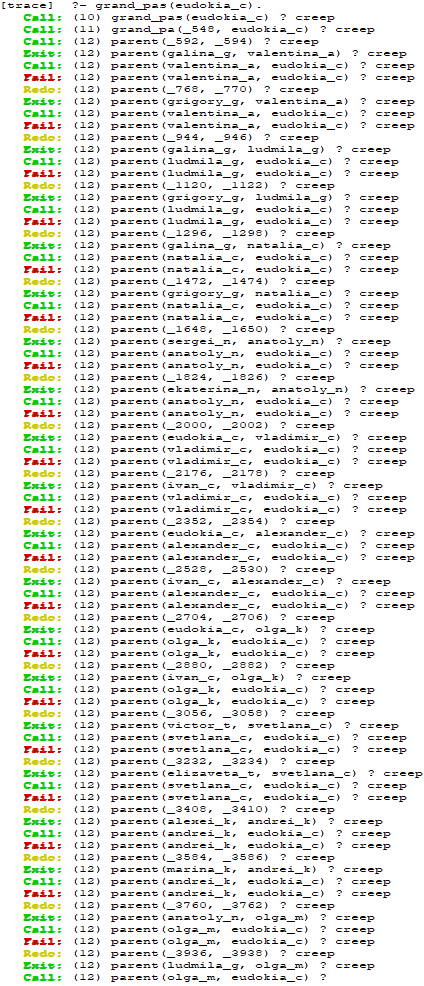
Пример 1.

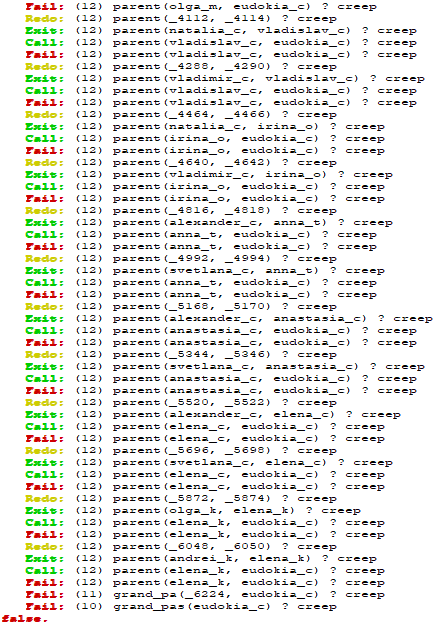




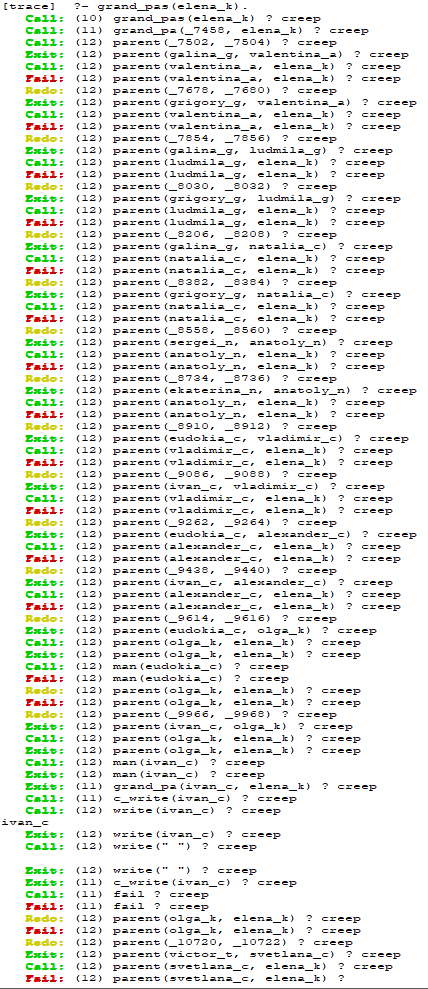


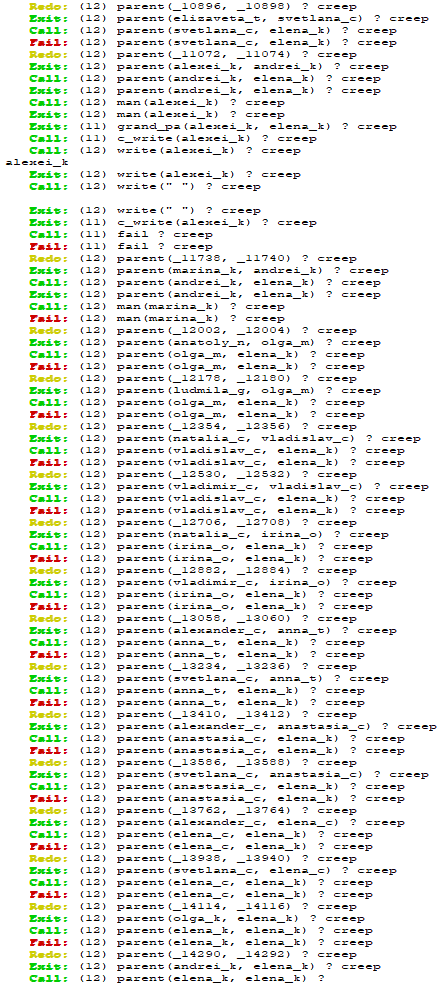
Пример 2.





Пример 3.







**Задание 10.**

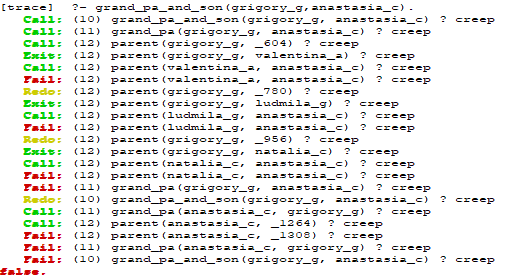
Построить предикат grand\_pa\_and\_son(X,Y), который проверяет, являются ли X и Y дедушкой и внуком или внуком и дедушкой. Провести трассировку последнего предиката, добавить снимок экрана в отчет для трех разных запросов к терминалу.

**Код:**

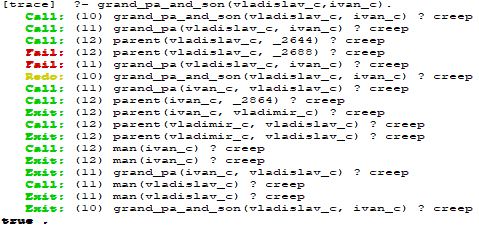
grand\_pa\_and\_son(X,Y):-( grand\_pa(X,Y),man(Y) )|( grand\_pa(Y,X),man(X) ).

**Скрины:**

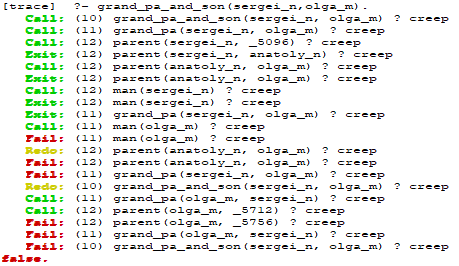
Пример 1.



Пример 2.



Пример 3.



**Задание 12.**

Построить предикат, который проверяет, является ли X дядей Y. Построить предикат, который выводит всех дядей X. Провести трассировку последнего предиката, добавить снимок экрана в отчет для трех разных запросов к терминалу.

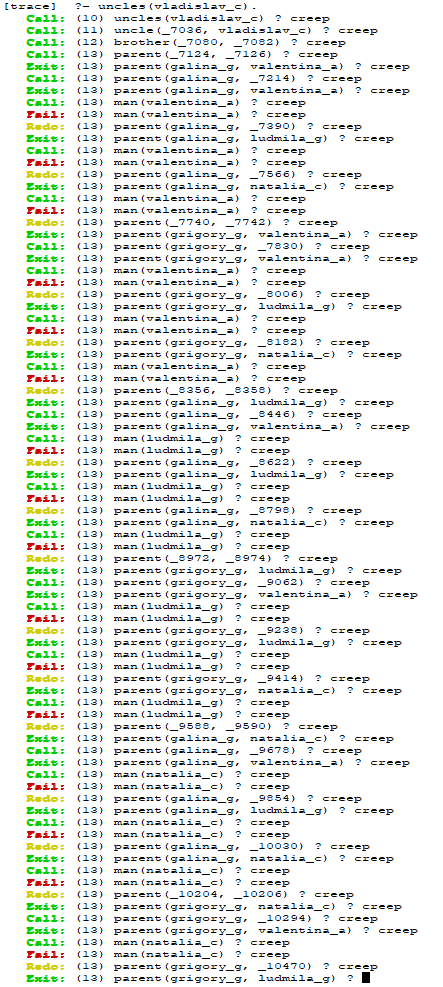
**Код:**

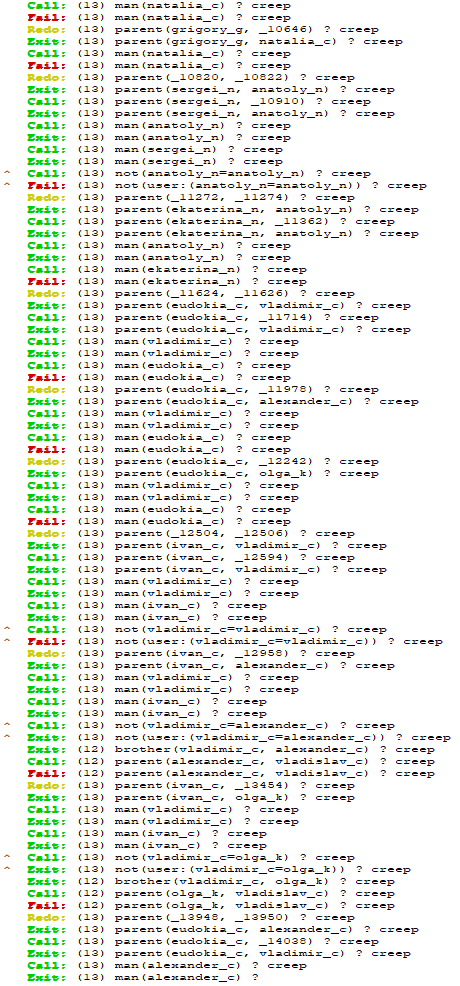
uncle(X,Y):-brother(X,A),parent(A,Y).

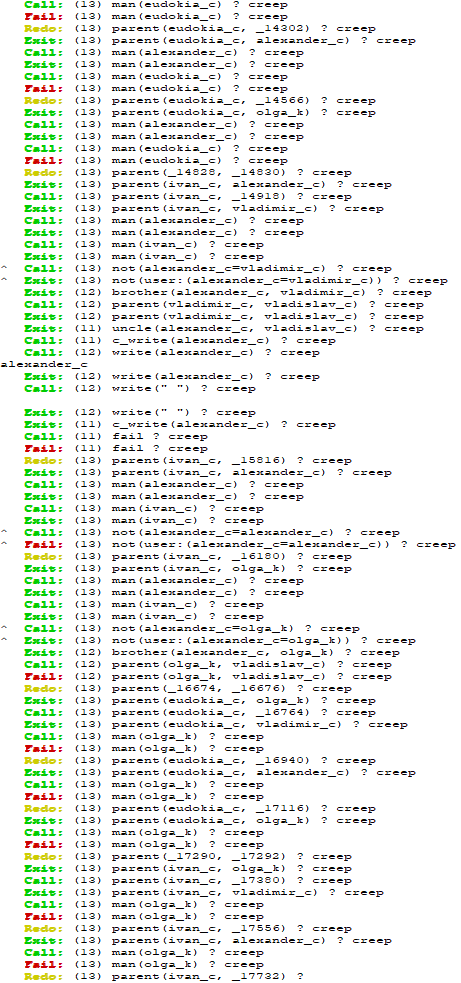
uncles(X):-uncle(Y,X),c\_write(Y),fail.

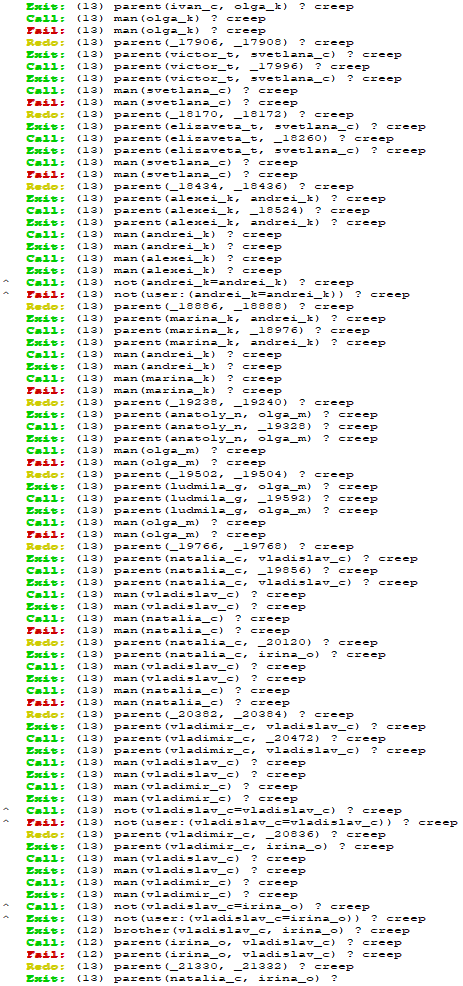
**Скрины:**

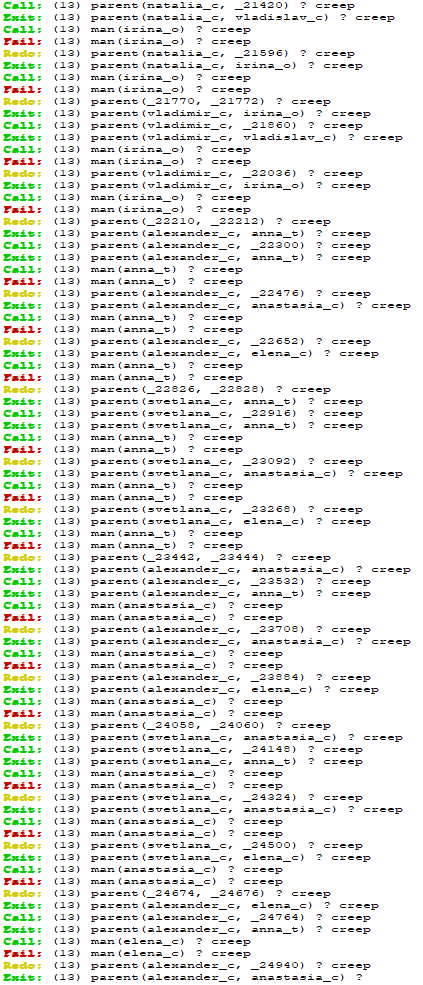
Пример 1.

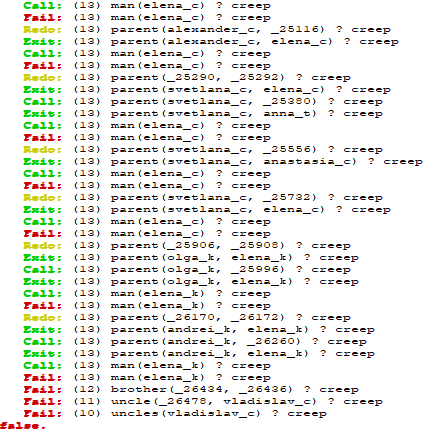




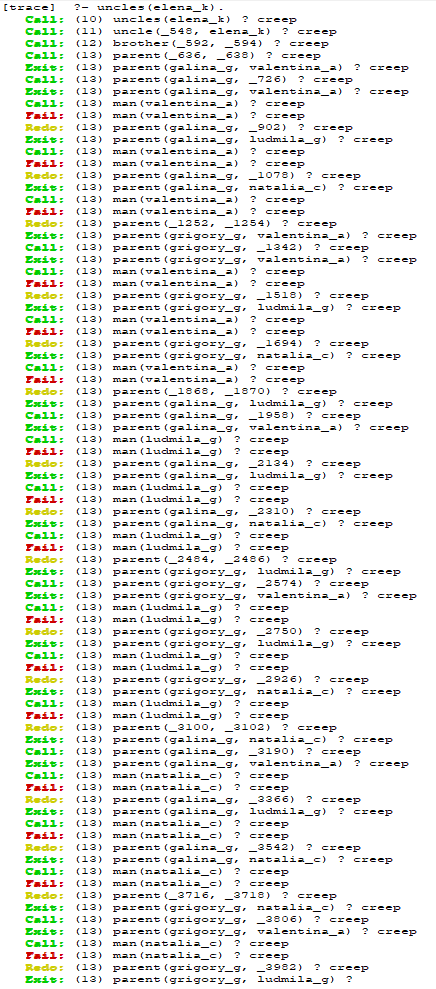


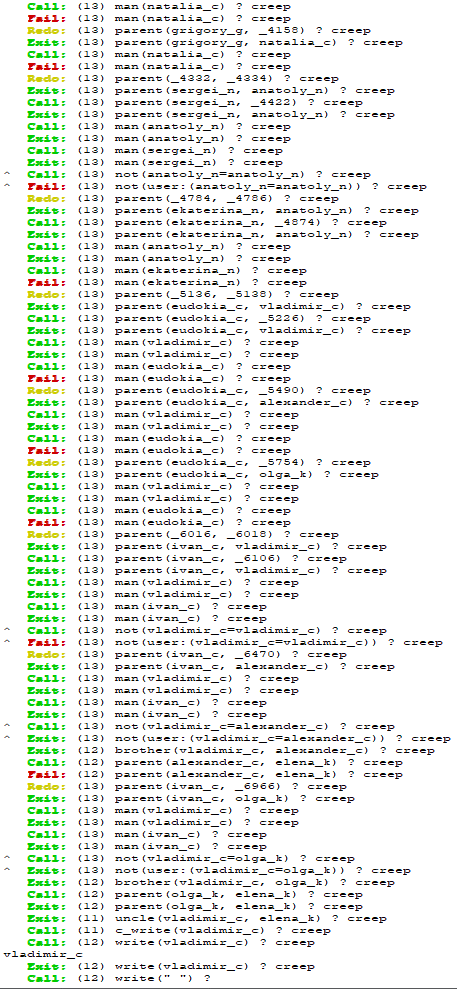


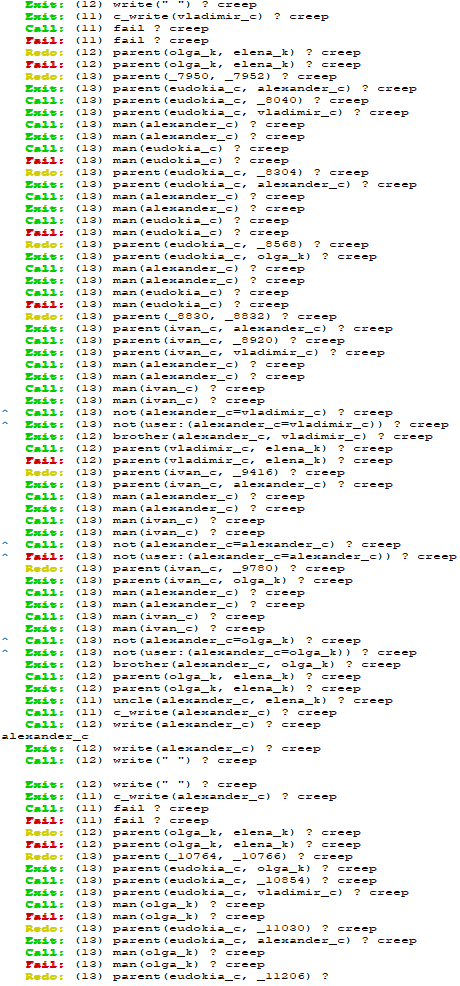


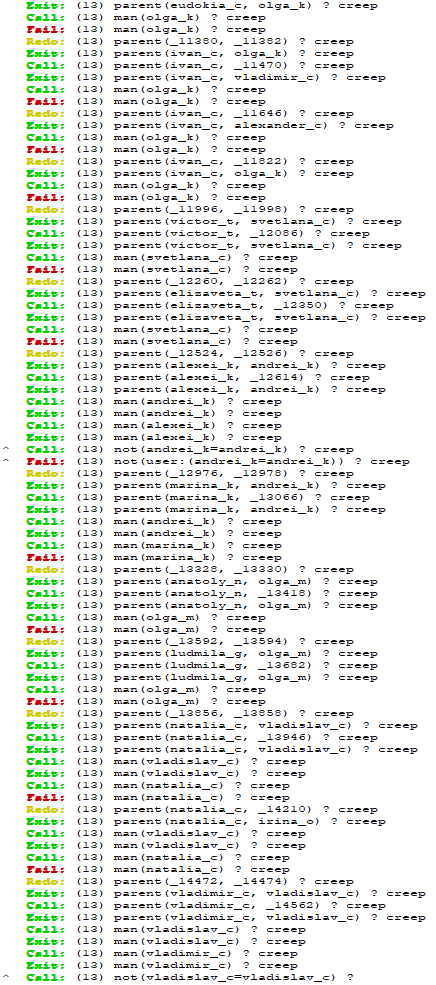


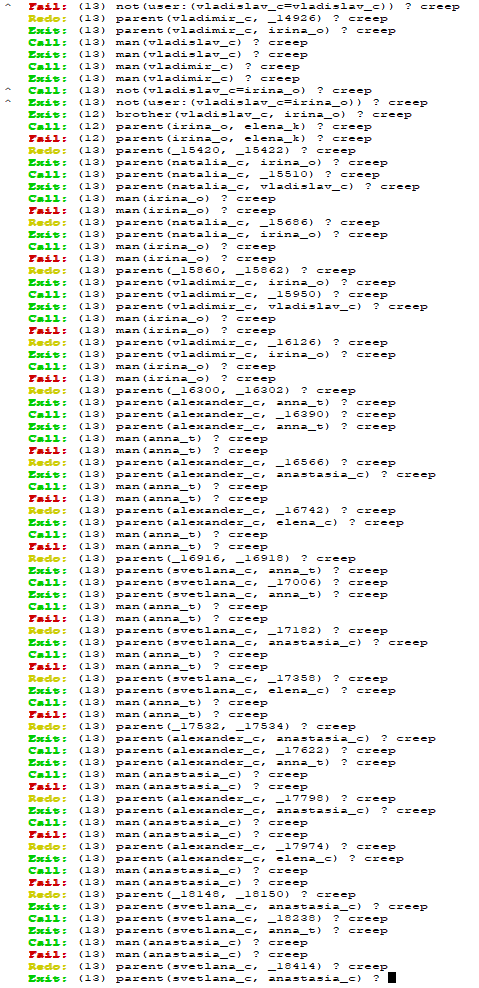
Пример 2.

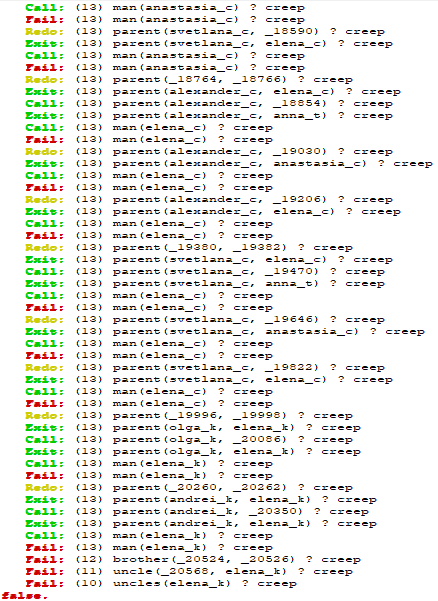












Пример 3.

