

Лабораторная работа №0

Задание. Создать свою базу фактов – своя родословная.

Листинг кода:

```
parent('Иван Кузьменко', 'Иван Кузьменко').
parent('Мария Кузьменко', 'Иван Кузьменко').
parent('Иван Кузьменко', 'Юлия Кузьменко').
parent('Анна Борисенко', 'Юлия Кузьменко').
parent('Юлия Кузьменко', 'Мария Кравченко').
parent('Остап Кравченко', 'Мария Кравченко').
parent('Василий Кравченко', 'Иван Кравченко').
parent('Варвара Кузьмук', 'Анна Кравченко').
parent('Иван Кравченко', 'Татьяна Кравченко').
parent('Мария Кравченко', 'Татьяна Кравченко').
parent('Татьяна Кравченко', 'Абрамов Максим').
parent('Вадим Абрамов', 'Абрамов Максим').

ancestor(A, B) :- parent(A, B).
ancestor(A, B) :- parent(A, C), ancestor(C, B).
ancestor(A, B) :- parent(A, B).
ancestor(A, B) :- parent(A, D), ancestor(D, B).
ancestor(A, B) :- parent(A, B).
ancestor(A, B) :- parent(A, E), ancestor(E, B).

?- writeln('Прадедушка Абрамова Максима:'),
   ancestor(A, 'Иван Кравченко'),
   writeln(A).

?- writeln('Дедушка и бабушка Абрамова Максима:'),
   ancestor(A, 'Татьяна Кравченко'),
   writeln(A),
   fail.

?- writeln('Родители Абрамова Максима:'),
   ancestor(A, 'Абрамов Максим'),
   writeln(A),
   fail;
true.
```

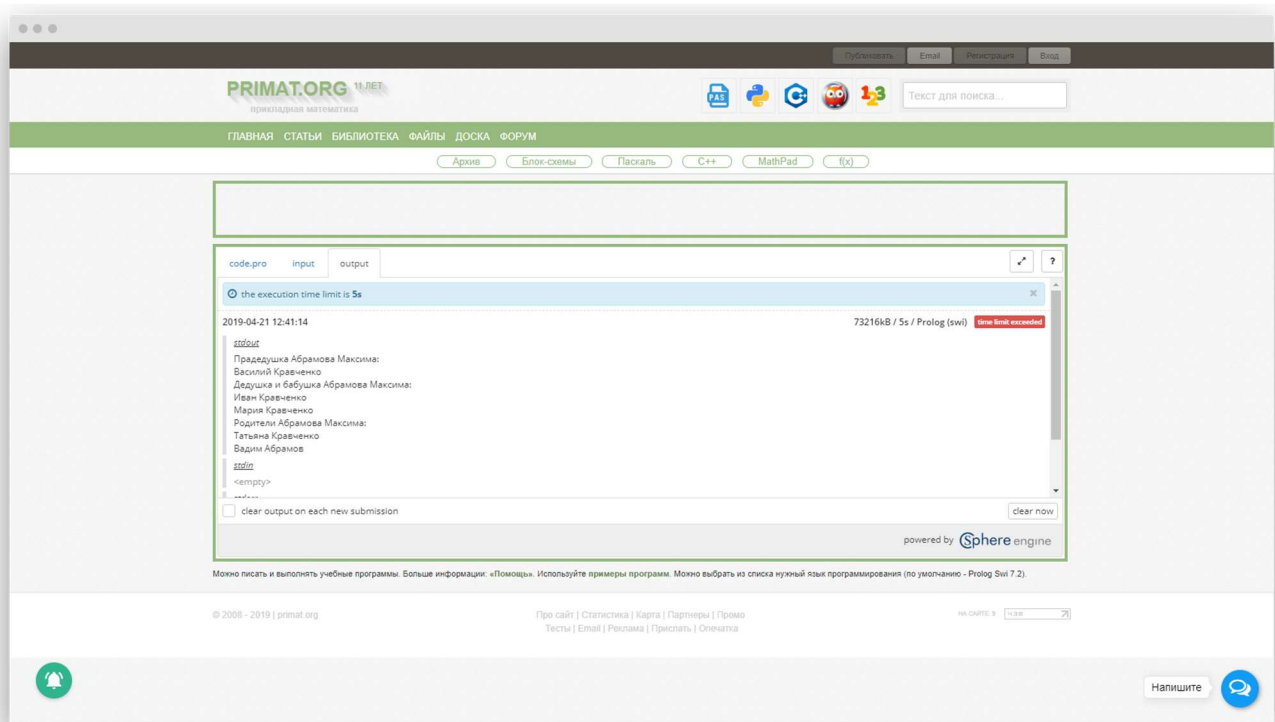


Рисунок 0.1 – Демонстрация работы программы

Лабораторна робота № 1

Тема. Ознайомлення з Visual Prolog

Мета: ознайомитися з інтерфейсом Visual Prolog, навчитися створювати проекти та запускати їх.

Завдання 1

1. Маємо базу даних, яка містить такі факти:

```
likes("Anna", apples).  
likes("Sergiy", bananas).  
likes("Andriy", apples).  
likes("Sveta", chocolate).  
likes("Vova", chocolate).  
likes("Anna", chocolate).  
likes("Sveta", oranges).  
likes("Vova", bananas).
```

Скласти програму, що визначає:

а) усіх, хто любить банани (bananas);

```
likes('Anna', apples).  
likes('Sergiy', bananas).  
likes('Andriy', apples).  
likes('Sveta', chocolate).  
likes('Vova', chocolate).  
likes('Anna', chocolate).  
likes('Sveta', oranges).  
likes('Vova', bananas).  
  
banana(A, B) :- likes(A, B).  
banana(A, B) :- likes(A, C), banana(C, B).  
  
?- writeln('Полюбляють банани:'),  
banana(A, bananas),  
writeln(A),  
fail;  
true.
```

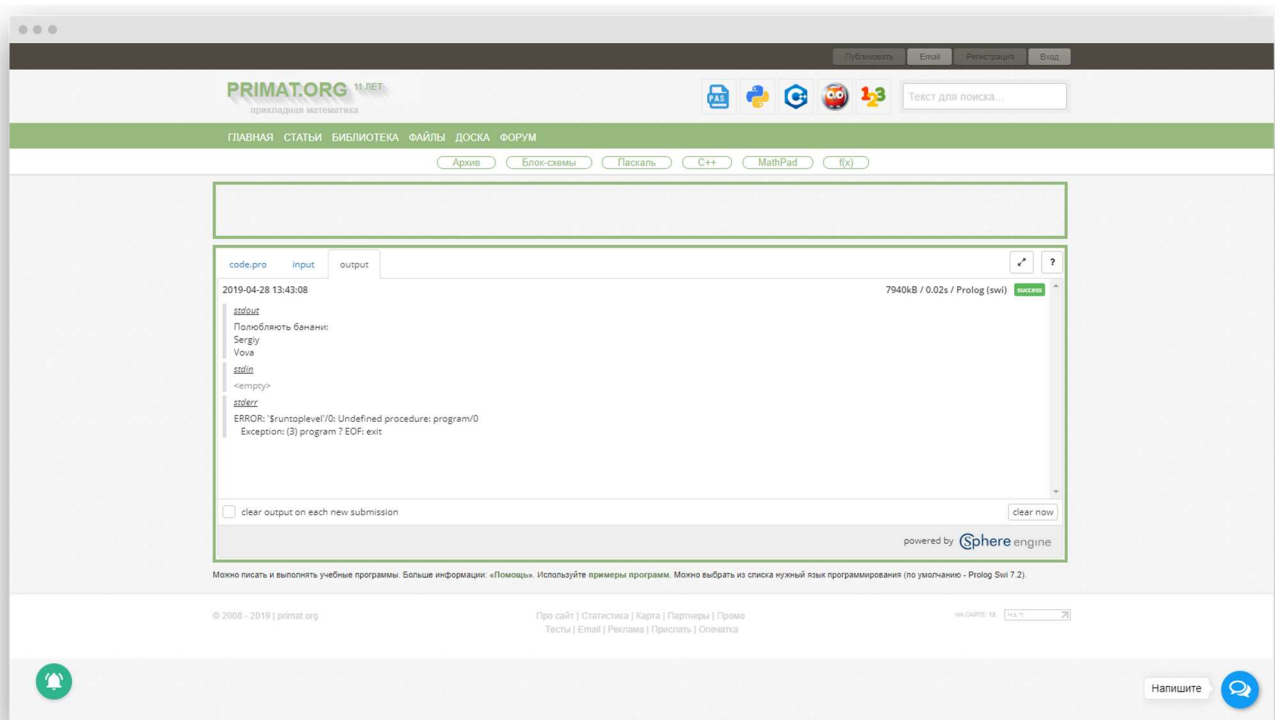


Рисунок 1.1.1а – демонстрація роботи програми

б) хто полюбляє і шоколад (chocolate), і яблука (apples);

```
likes('Anna', apples).  
likes('Sergiy', bananas).  
likes('Andriy', apples).  
likes('Sveta', chocolate).  
likes('Vova', chocolate).  
likes('Anna', chocolate).  
likes('Sveta', oranges).  
likes('Vova', bananas).
```

```
choco(A, B) :- likes(A, B).  
choco(A, B) :- likes(A, C), choco(C, B).  
app(A, B) :- likes(A, B).  
app(A, B) :- likes(A, D), app(D, B).
```

```
?- writeln('Полюбляють шоколад:'),  
choco(A, chocolate),  
writeln(A),  
fail;  
true.  
  
?- writeln('Полюбляють яблука:'),  
app(A, apples),  
writeln(A),  
fail;  
true.
```

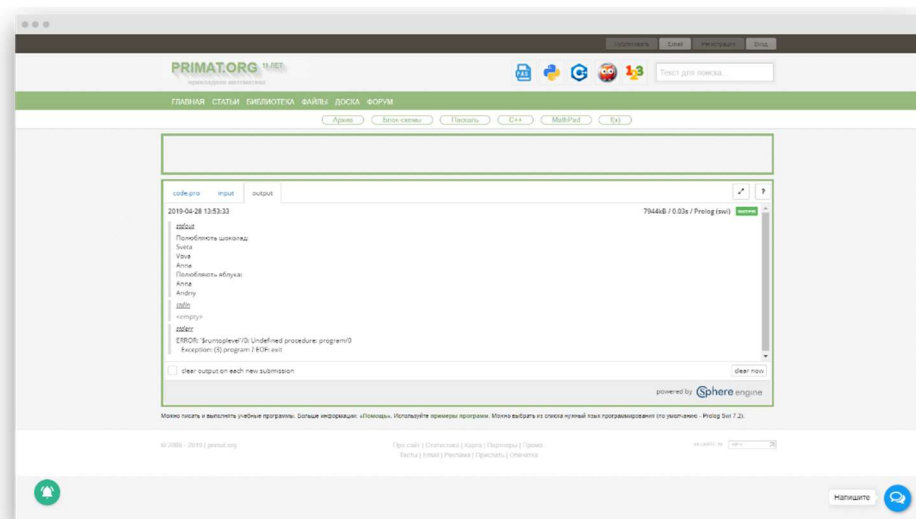


Рисунок 1.1.16 – демонстрація роботи програми

в) що любить Вова;

```
likes('Anna', apples).  
likes('Sergiy', bananas).  
likes('Andriy', apples).  
likes('Sveta', chocolate).  
likes('Vova', chocolate).  
likes('Anna', chocolate).  
likes('Sveta', oranges).  
likes('Vova', bananas).
```

```
vova(A, B) :- likes(A, B).  
vova(A, B) :- likes(C, B), vova(A, C).
```

```
?- writeln('Що любить Вова:'),  
vova('Vova', B),  
writeln(B),  
fail;  
true.
```

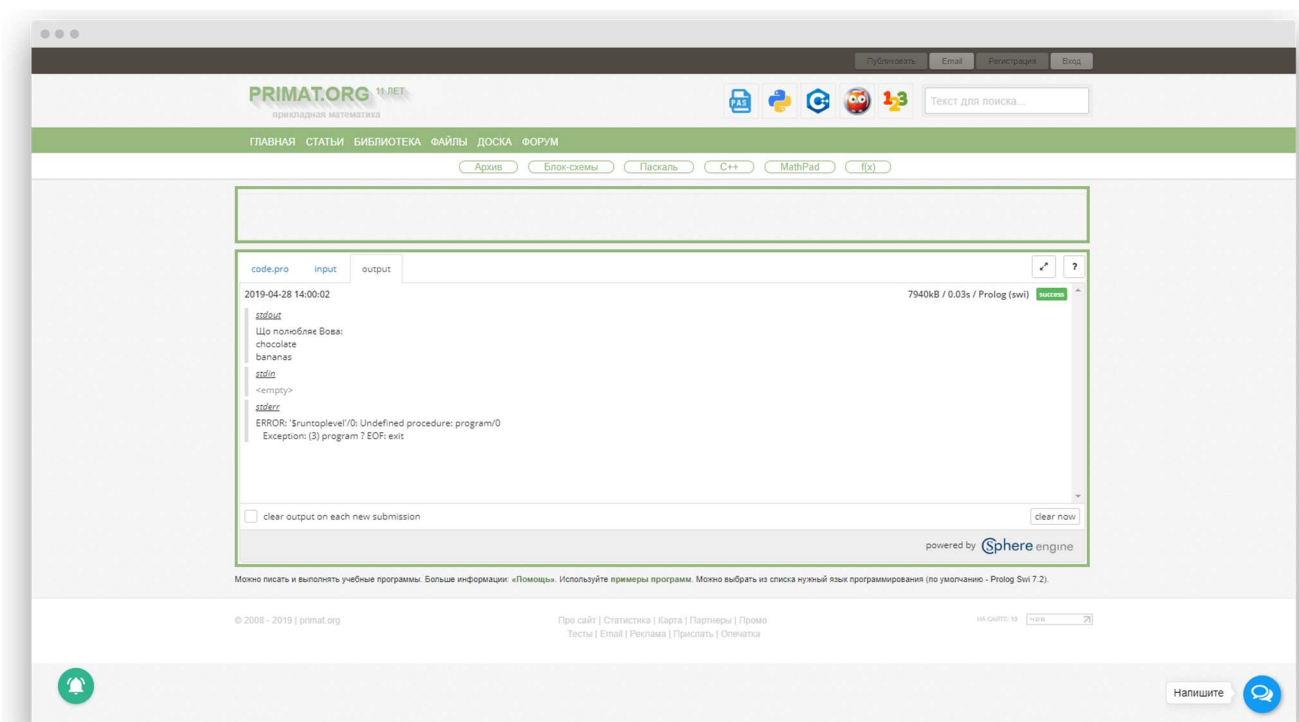


Рисунок 1.1.1в – демонстрація роботи програми

г) що люблять і Свєта, і Вова.

```
likes('Anna', apples).
likes('Sergiy', bananas).
likes('Andriy', apples).
likes('Sveta', chocolate).
likes('Vova', chocolate).
likes('Anna', chocolate).
likes('Sveta', oranges).
likes('Vova', bananas).

svet(A, B) :- likes(A, B).
svet(A, B) :- likes(C, B), svet(A, C).
vova(A, B) :- likes(A, B).
vova(A, B) :- likes(D, B), vova(A, D).

?- writeln('Що любляє Свєта:'),
   svet('Sveta', B),
   writeln(B),
   fail;
true.

?- writeln('Що любляє Вова:'),
   vova('Vova', B),
   writeln(B),
   fail;
true.
```

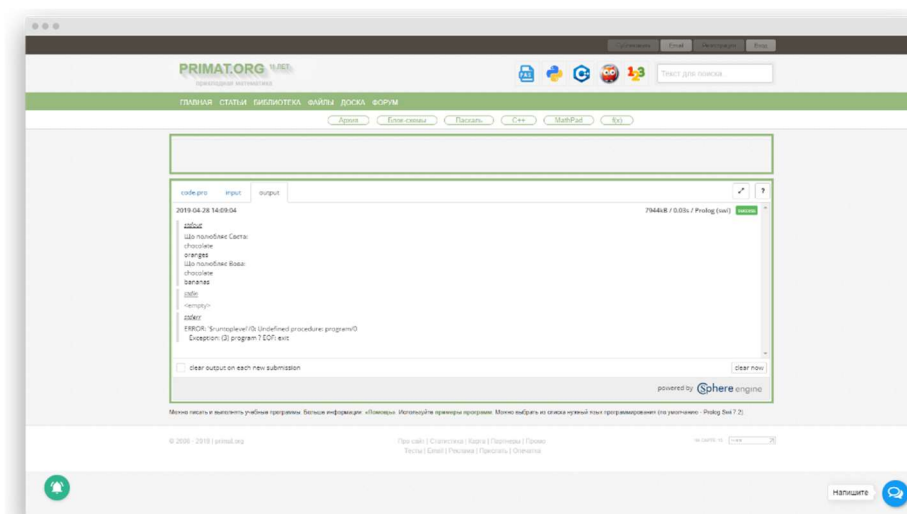


Рисунок 1.1.1г – демонстрація роботи програми

2. Маємо базу даних, що містить такі факти:

```
plays("Sasha", football).  
plays("Katya", tennis).  
plays("Sasha", tennis).  
plays("Andriy", football).  
plays("Oleg", football).  
plays("Olga", tennis).  
plays("Katya", volleyball).  
plays("Oleg", volleyball).
```

Скласти програму, що визначає:

а) яким видом спорту захоплюється Андрій;

```
plays('Sasha', football).  
plays('Katya', tennis).  
plays('Sasha', tennis).  
plays('Andriy', football).  
plays('Oleg', football).  
plays('Olga', tennis).  
plays('Katya', volleyball).  
plays('Oleg', volleyball).
```

```
svet(A, B) :- plays(A, B).  
svet(A, B) :- plays(C, B), svet(A, C).
```

```
?- writeln('Яким видом спорту захоплюється Андрій:'),  
svet('Andriy', B),  
writeln(B),  
fail;  
true.
```

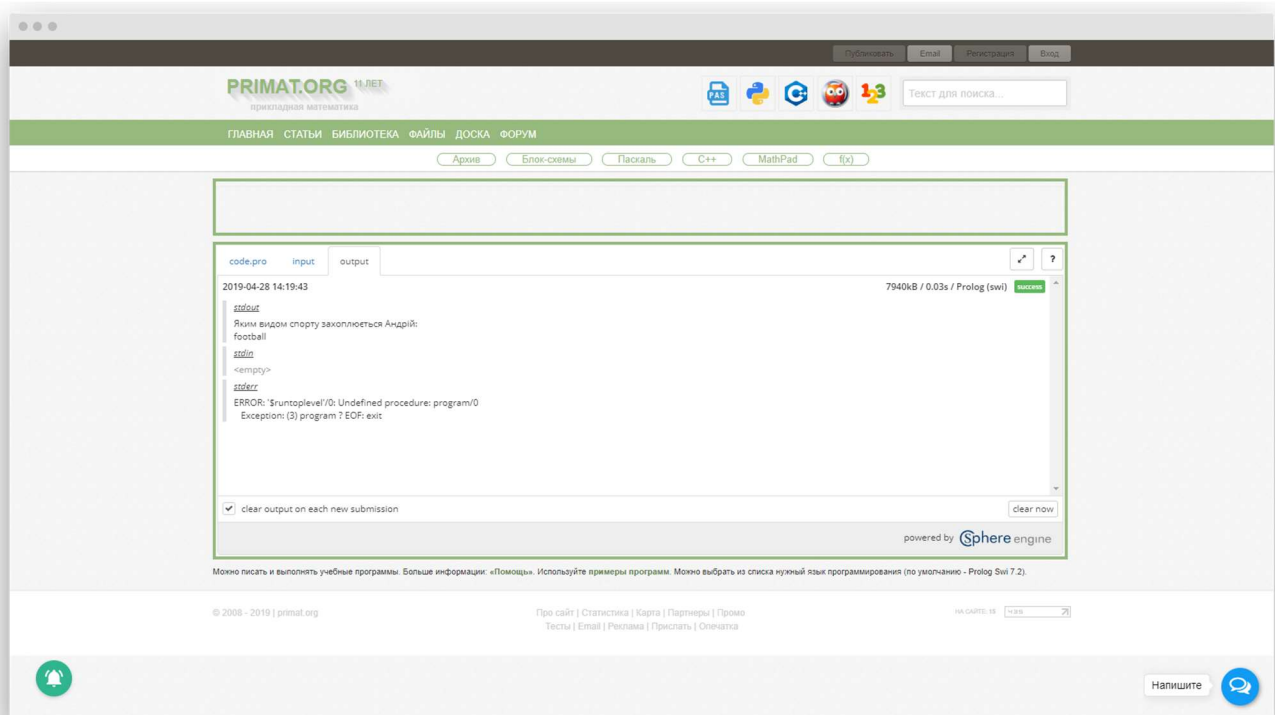


Рисунок 1.1.2a – демонстрація роботи програми

б) усіх, хто грає у волейбол;

```
plays('Sasha', football).
plays('Katya', tennis).
plays('Sasha', tennis).
plays('Andriy', football).
plays('Oleg', football).
plays('Olga', tennis).
plays('Katya', volleyball).
plays('Oleg', volleyball).
```

```
svet(A, B) :- plays(A, B).
svet(A, B) :- plays(A, C), svet(C, B).
```

```
?- writeln('Усі, хто грає у волейбол:'),
svet(A, volleyball),
writeln(A),
fail;
true.
```

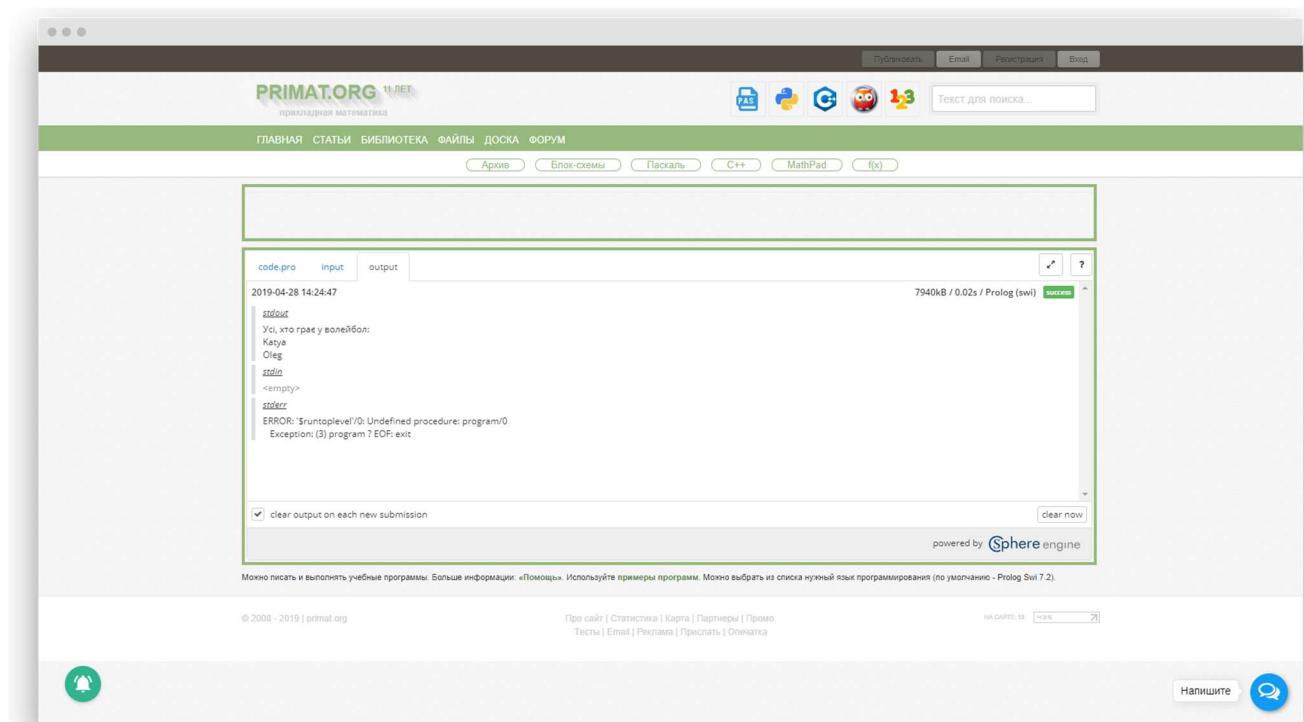


Рисунок 1.1.26 – демонстрація роботи програми

в) яким видом спорту захоплюються і Ольга, і Саша;

```
plays('Sasha', football).
plays('Katya', tennis).
plays('Sasha', tennis).
plays('Andriy', football).
plays('Oleg', football).
plays('Olga', tennis).
plays('Katya', volleyball).
plays('Oleg', volleyball).
```

```
svet(A, B) :- plays(A, B).
svet(A, B) :- plays(C, B), svet(A, C).
vova(A, B) :- plays(A, B).
vova(A, B) :- plays(D, B), vova(A, D).
```



```

?- writeln('Яким видом спорту захоплюються Ольга:'),
vova('Olga', B),
writeln(B),
fail;
true.
?- writeln('Яким видом спорту захоплюються Саша:'),
vova('Sasha', B),
writeln(B),
fail;
true.

```

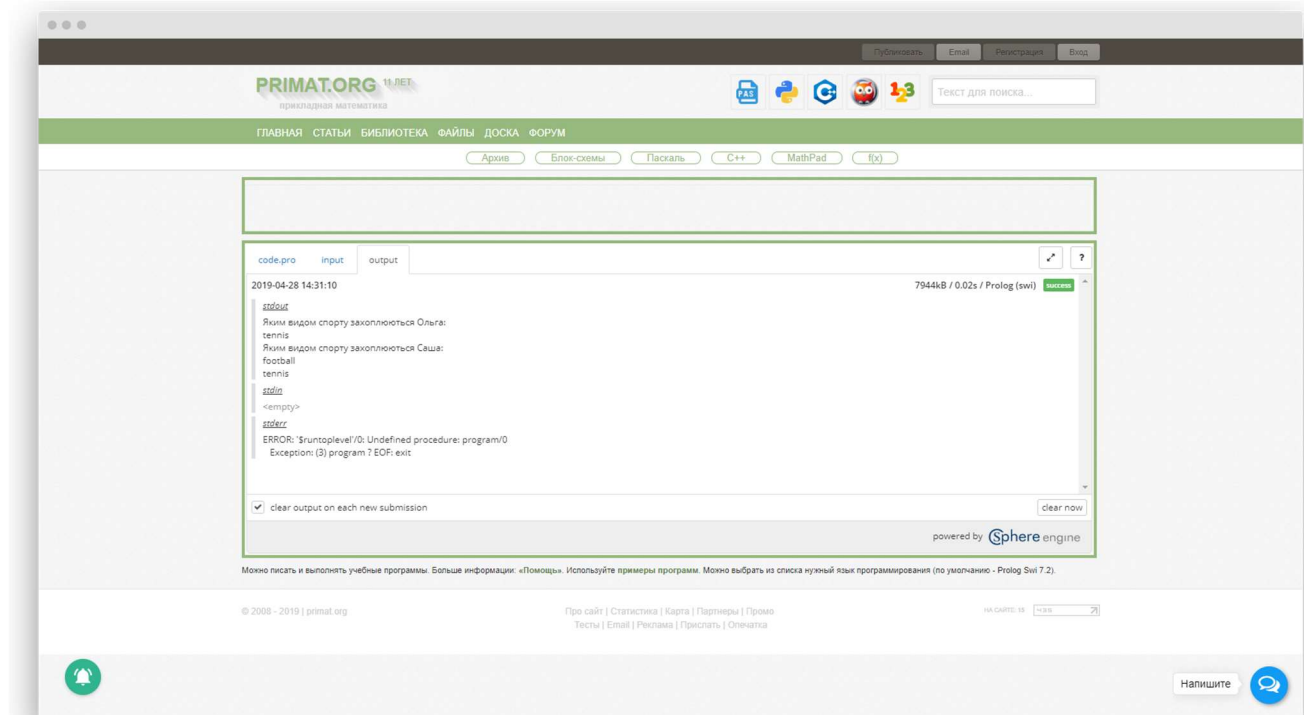


Рисунок 1.1.2в – демонстрація роботи програми

г) хто захоплюється і футболом, і волейболом.

```

plays('Sasha', football).
plays('Katya', tennis).
plays('Sasha', tennis).
plays('Andriy', football).
plays('Oleg', football).
plays('Olga', tennis).
plays('Katya', volleyball).
plays('Oleg', volleyball).

```

```

svet(A, B) :- plays(A, B).
svet(A, B) :- plays(A, C), vova(C, B).
vova(A, B) :- plays(A, B).
vova(A, B) :- plays(A, D), vova(D, B).

```

```

?- writeln('Хто захоплюється футболлом:'),
vova(A, football),
writeln(A),
fail;
true.

?- writeln('Хто захоплюється волейболом:'),
vova(A, volleyball),
writeln(A),
fail;
true.

```

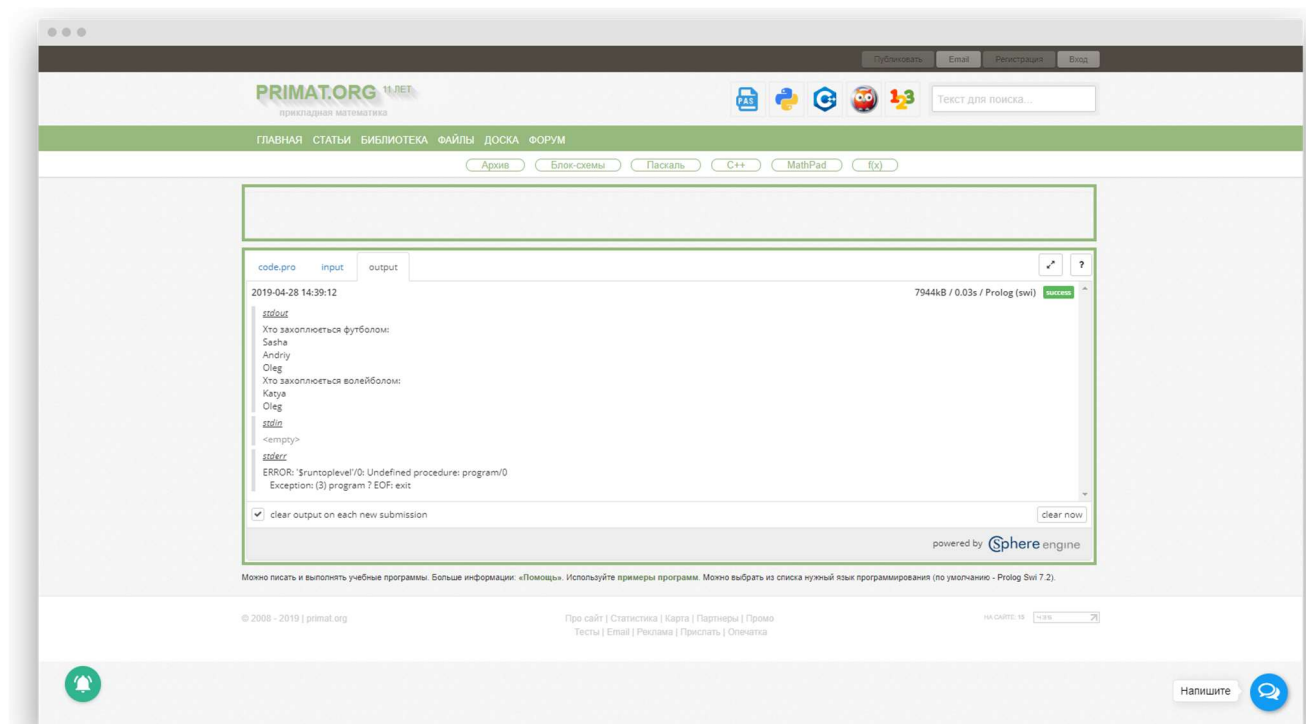


Рисунок 1.1.2г – демонстрація роботи програми

3. Маємо базу даних, що містить такі факти:

```

likes("Anna", apples).
likes("Sergiy", bananas).
likes("Andriy", apples).
likes("Sveta", chocolate).
likes("Vova", chocolate).
likes("Anna", chocolate).
likes("Sveta", oranges).
likes("Vova", bananas).
fruits(apples).
fruits(bananas).
fruits(oranges).
candies(chocolate).

```

Використовуючи наявні факти, скласти нове правило *likes_fruits(X)* (полюбляє фрукти) і визначити всіх, хто полюбляє фрукти;

```
likes('Anna', apples).
likes('Sergiy', bananas).
likes('Andriy', apples).
likes('Sveta', chocolate).
likes('Vova', chocolate).
likes('Anna', chocolate).
likes('Sveta', oranges).
likes('Vova', bananas).
fruits(apples).
fruits(bananas).
fruits(oranges).
candies(chocolate).

likes_fruits(A, B) :- likes(A, B).
likes_fruits(A, B) :- likes(A, X), fruits(X).

?- writeln('Хто полюбляє фрукти:'),
likes_fruits(A, fruits),
writeln(A),
fail;
true.
```

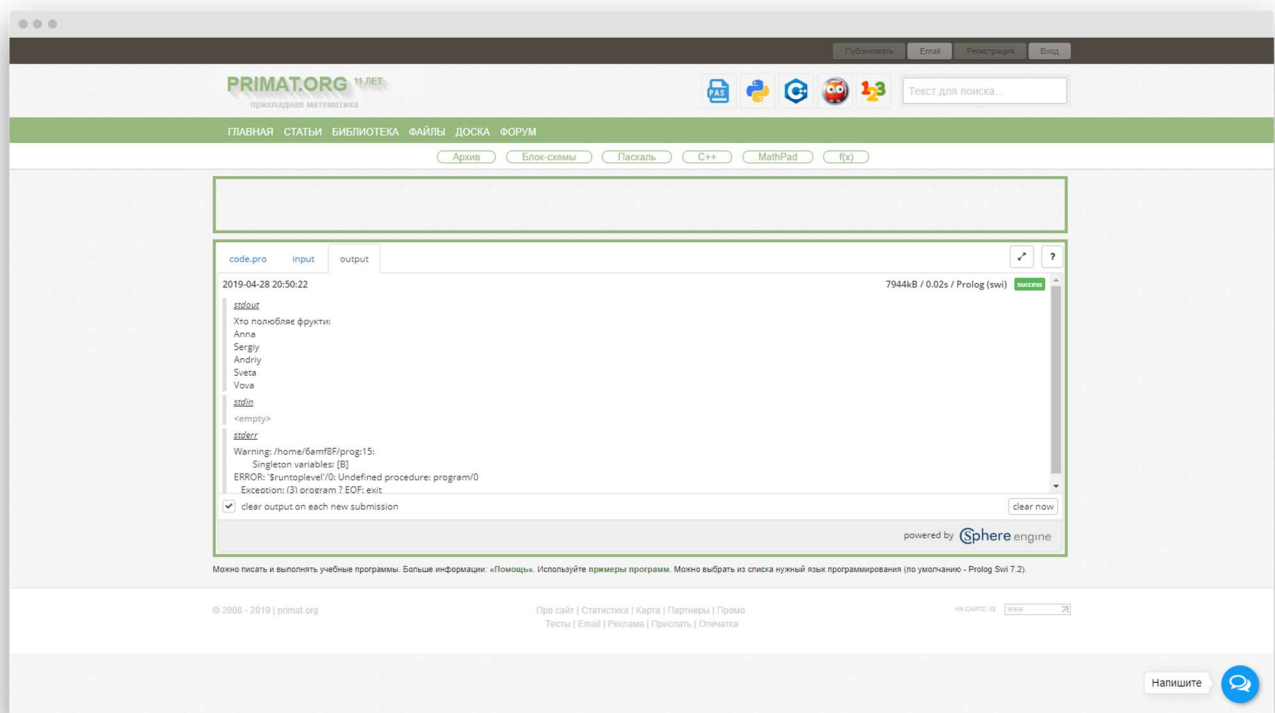


Рисунок 1.1.3 – демонстрація роботи програми

4. Маємо базу даних, що містить такі факти:

```
plays("Sasha", football).
plays("Katya", tennis).
plays("Sasha", tennis).
plays("Andriy", football).
plays("Oleg", football).
plays("Olga", tennis).
plays("Katya", volleyball).
woman("Katya").
woman("Olga").
man("Sasha").
man("Andriy").
man("Oleg").
```

Використовуючи наявні факти, скласти нове правило *volleyball_woman(X)* і визначити всіх жінок, що грають у волейбол.

```
plays('Sasha', football).
plays('Katya', tennis).
plays('Sasha', tennis).
plays('Andriy', football).
plays('Oleg', football).
plays('Olga', tennis).
plays('Katya', volleyball).
woman('Katya').
woman('Olga').
man('Sasha').
man('Andriy').
man('Oleg').
```

```
volleyball_woman(A, B) :- plays(A, B).
volleyball_woman(A, B) :- plays(A, X), woman(X).
```

```
?- writeln('Всі жінок, що грають у волейбол:'),
volleyball_woman(A, woman),
writeln(A),
fail;
true.
```

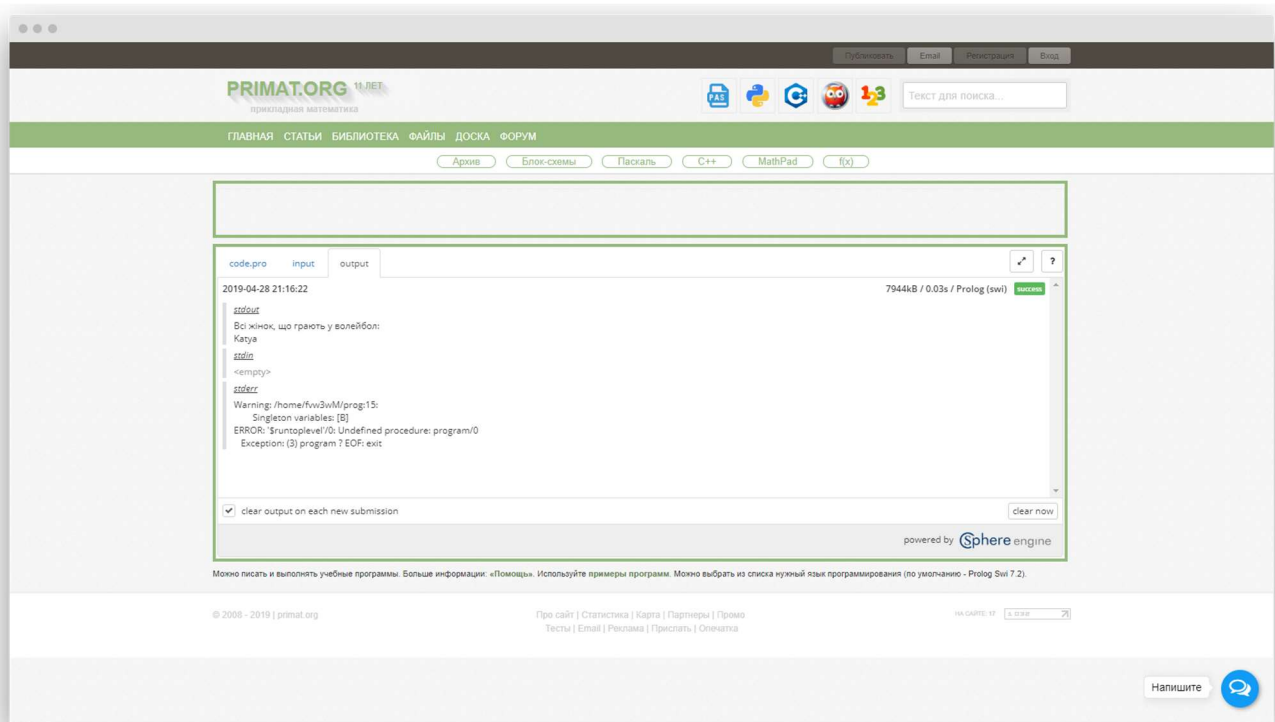


Рисунок 1.1.4 – демонстрація роботи програми

Завдання 2

Допрацюйте вихідний код прикладу таким чином: додайте нове правило *grandmother* (бабуся) і визначте, хто є бабусею.

```
parent('Ivan', 'Katya').
parent('Anna', 'Oleg').
parent('Oleg', 'Dima').
parent('Igor', 'Olga').
parent('Oleg', 'Viktor').
parent('Igor', 'Ivan').
man('Dima').
man('Ivan').
man('Igor').
man('Oleg').
man('Viktor').
woman('Katya').
woman('Olga').
woman('Anna').
```

```
grandmother(X,Z):-parent(X,Y), parent(Y,Z), woman(X).
```

```
?- writeln('бабуся:'),
grandmother(X, 'Dima'),
writeln(X),
fail;
true.
```

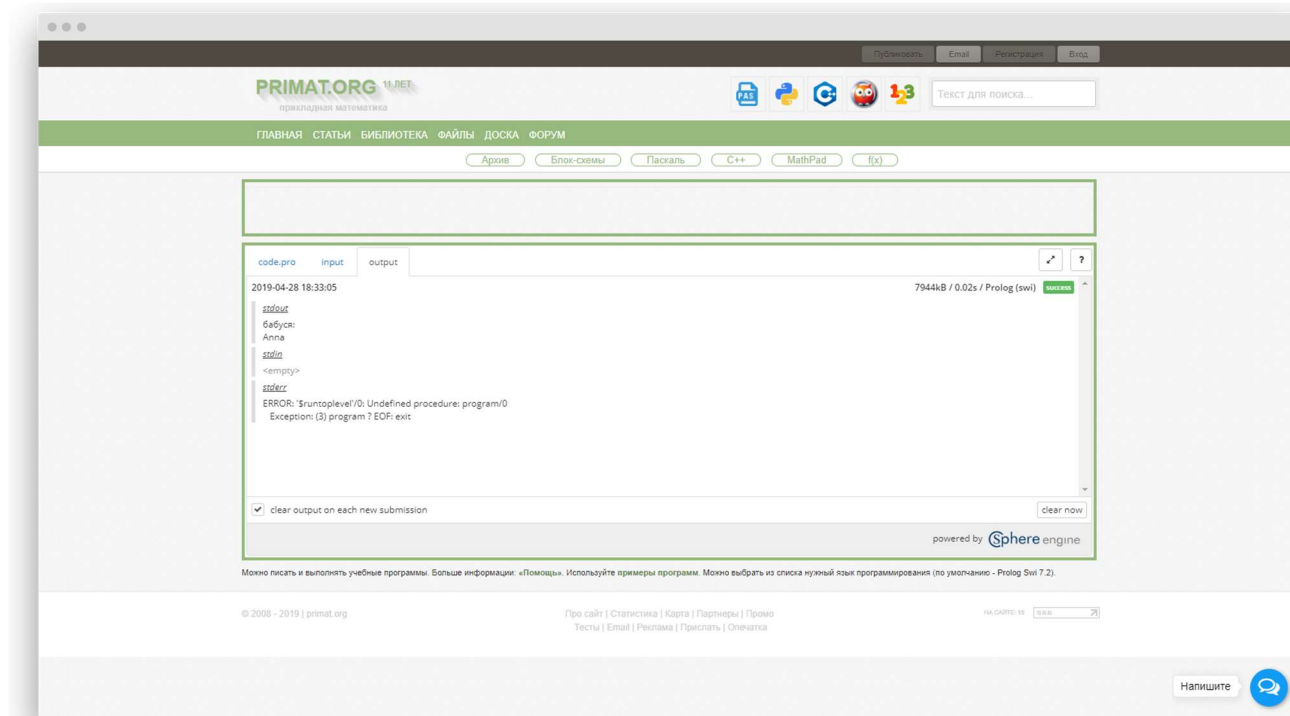


Рисунок 1.2.1 – демонстрація роботи програми

Контрольні запитання

1. З чого складається логічна модель знань?

Логічна модель представлення знань — модель у представленні знань.

Основна ідея підходу при побудові логічних моделей представлення знань — вся інформація, необхідна для вирішення прикладних завдань, розглядається як сукупність фактів і тверджень, які представляються як формули в деякій логіці. Знання відображаються сукупністю таких формул, а отримання нових знань зводиться до реалізації процедур логічного висновку. У основі логічних моделей представлення знань лежить поняття формальної теорії, кортеж

$S = \langle B, F, A, R \rangle$, де:

B — зліченна множина базових символів (алфавіт);

F — множина, елементи якої називають формулами;

A — виділена підмножина апіорі справедливих формул (аксіом);

R — зліченна множина відношень між формулами, яку називають правилами висновку.

2. Які основні розділи Visual Prolog-програми?

Програма на Visual Prolog складається з наступних основних розділів:

директиви компілятора;

CONSTANTS - опис констант;

DOMAINS - опис доменів;

FACTS - опис предикатів внутрішньої бази даних;

PREDICATES - опис предикатів;

CLAUSES - опис тверджень;

GOAL - опис внутрішньої мети.

3. Що називається правилом та фактом на мові Visual Prolog?

Факти використовуються для представлення даних, а **правила** використовуються для виведення логічних висновків, тобто нових фактів. Факти і правила називаються твердженнями. Типова програма на Пролозі містить факти і правила, засновані на самих різних взаємозв'язках предикатів.

4. Що таке база знань?

База знань — це особливого роду база даних, розроблена для управління знаннями (метаданими), тобто збором, зберіганням, пошуком і видачею знань. Розділ штучного інтелекту, що вивчає бази даних і методи роботи із знаннями, називається інженерією знань

5. Що таке предикат?

Предикат у сучасній логіці зазвичай означає булевозначну функцію $P: X \rightarrow \{0, 1\}$, яка називається предикатом на X . Однак, предикати мають багато різних інтерпретацій та способів використання у математиці та логіці, і їх точне визначення різниться від теорії до теорії.

6. Для чого потрібні змінні в Visual Prolog?

Змінними є всі ідентифікатори що починаються з великої букви. Але окрім звичайних змінних, є змінна що позначається "_", і називається анонімною. Якщо вживати її в запиті, це означатиме, що значення тієї змінної нас не цікавить.

Лабораторна робота № 2

Тема. Пошук з поверненням. Керування пошуком з поверненням:

предикати fail та відтинання

Мета: ознайомитися з організацією та керуванням пошуку з поверненням, предикатами fail та відтинанням.

Завдання 1

1. База даних містить такі факти:

```
interested("Kolya", guitar).
interested("Olya", violin).
interested("Dima", swimming).
interested("Tanya", tennis).
sport(swimming).
sport(tennis).
mus_instr(violin).
mus_instr(guitar).
```

Створити правило sportsman (спортсмен) і визначити, хто захоплюється спортом;

```
interested('Kolya', guitar).
interested('Olya', violin).
interested('Dima', swimming).
interested('Tanya', tennis).
sport(swimming).
sport(tennis).
mus_instr(violin).
mus_instr(guitar).

sportsman(A, B) :- interested(A, B).
sportsman(A, B) :- interested(A, X), sport(X).

?- writeln('Спортсмены:'),
   sportsman(A, sport),
   writeln(A),
   fail;
true.
```

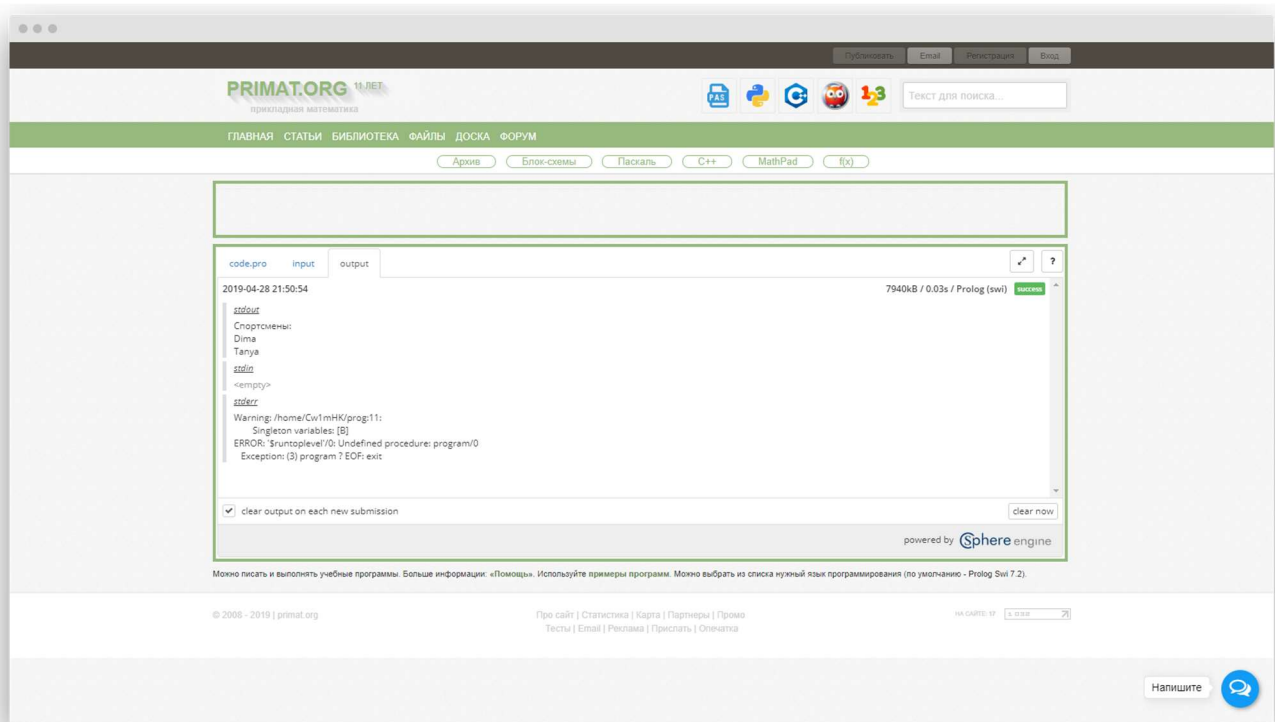



Рисунок 2.1.1 – демонстрація роботи програми

2. База даних містить такі факти:

```
interested("Dima", swimming).
interested("Tanya", tennis).
interested("Kolya", guitar).
interested("Olya", violin).
sport(swimming).
sport(tennis).
mus_instr(violin).
mus_instr(guitar).
```

Створити правило musician (музикант) і визначити, хто захоплюється музикою;

```
interested('Kolya', guitar).
interested('Olya', violin).
interested('Dima', swimming).
interested('Tanya', tennis).
sport(swimming).
sport(tennis).
mus_instr(violin).
mus_instr(guitar).
```

```
musician(A, B) :- interested(A, B).
musician(A, B) :- interested(A, X), mus_instr(X).
```

```
?- writeln('Музыканты:'),
musician(A, mus_instr),
writeln(A),
fail;
true.
```

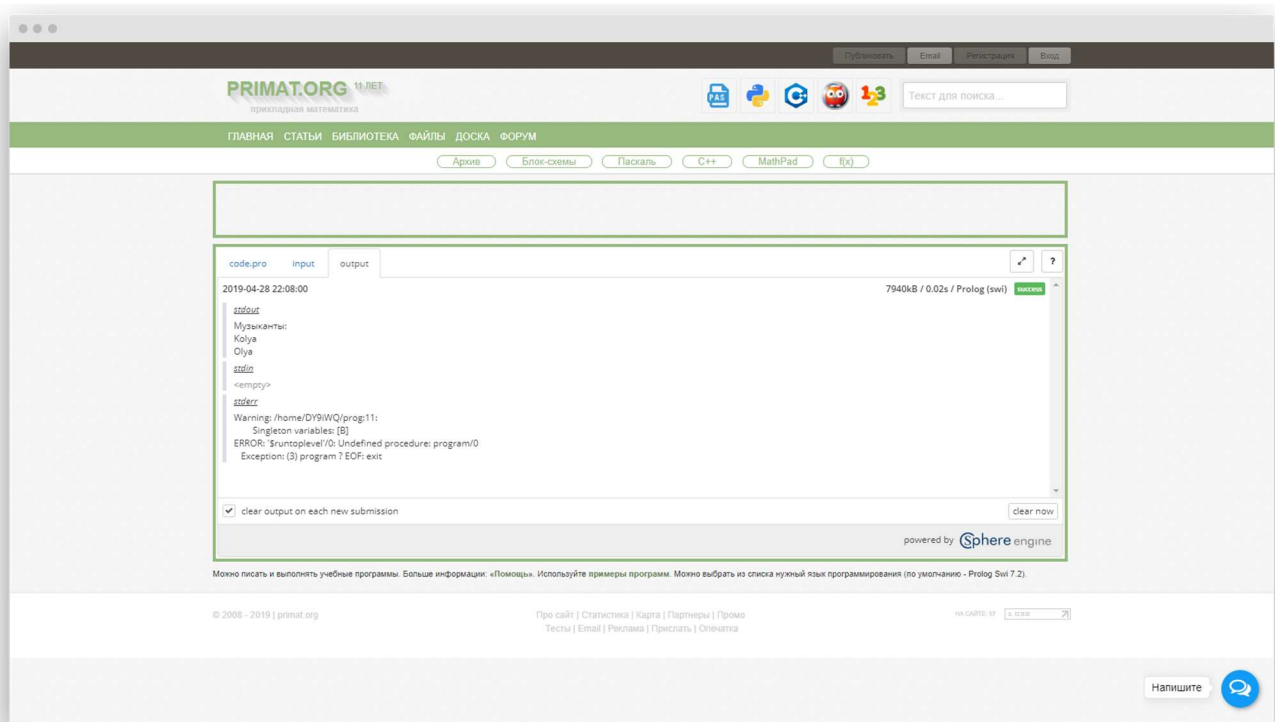


Рисунок 2.1.2 – демонстрація роботи програми

Завдання 2

База даних містить факти виду: `rest(name, city)`, `ukraine(city)`, `russia(city)`, `woman(name)`, `man(name)`.

Вивести список жінок, що відпочивають у Росії;

```

rest(name, city).
rest(nam, cit).
ukraine(cit).
russia(city).
woman(name).
man(nam).

musician(A, B) :- rest(A, B).
musician(A, B) :- rest(A, X), woman(X).

?- writeln('Женщины отдыхающие в РФ:'),
musician(A, city),
writeln(A),
fail;
true.

```

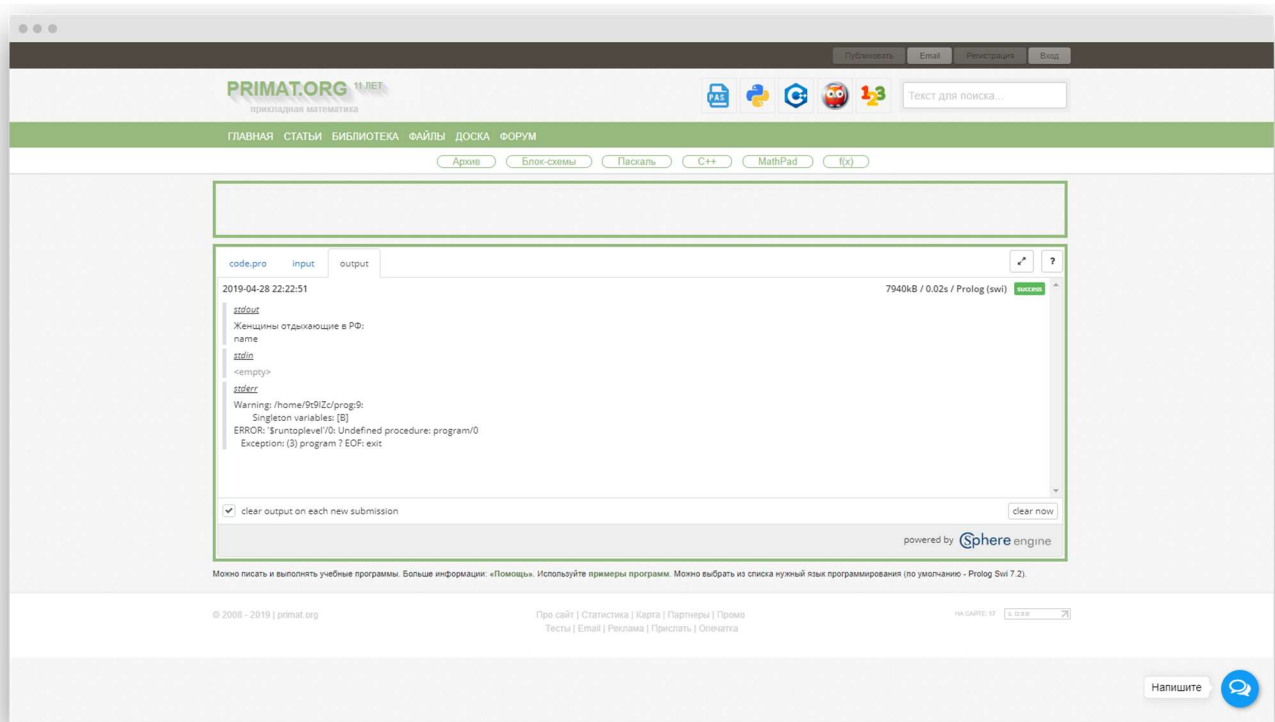


Рисунок 2.2.1 – демонстрація роботи програми

Контрольні питання

1. Для чого використовується предикат fail?

Fail - це невдача. Він здійснює відкат, так що предикати і правила виконуються ще раз.

2. Яку роль відіграє відтинання?

а) коли ви знаєте попередньо, що певні варіанти ніколи не дадуть поштовху в знаходженні розв'язку, тоді використання cut (зелений cut) відкидає перегляд альтернативних шляхів.

б) коли логіка програми потребує використання cut для відкидання перегляду альтернативних підцілей, тоді його називають червоним відтинанням.

3. Як запустити налаштовувальник покрокового виконання програми?

Крок виконання програми – це рядок програми. У рядку можуть бути записані одна або кілька команд програми.

4. Для чого використовується предикат readln?

Предикат вводу рядка.