

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Сибирский государственный университет телекоммуникаций и информатики»
(СибГУТИ)

Кафедра прикладной математики и кибернетики

РАСЧЁТНО-ГРАФИЧЕСКОЕ ЗАДАНИЕ
по дисциплине «Функциональное и логическое программирование»

Вариант 3

Выполнил:

студент группы ИС-241

Бондаренко А.А.
ФИО студента

Работу проверил: Сороковых Д.А.
ФИО преподавателя

Новосибирск 2024 г.

ЗАДАНИЕ

1. Сформируйте новый список из общих элементов двух списков (пересечение множеств).
Например, [1,2,3,5], [6,4,1,8,3]-> [1,3].
Списки должны вводиться с клавиатуры во время выполнения программы, использование рекурсии обязательно.
2. Определите, совпадают ли два текстовых файла. Если нет, то получите номер первой строки и позиции первого символа в этой строке, в которых файлы отличаются между собой.
Исходный файл должен содержать несколько строк.

ЛИСТИНГ

Первое задание:

```
1 | % Чтение строки и преобразование её в список чисел
2 | read_list(List) :-
3 |     read_line_to_string(user_input, Input),
4 |     split_string(Input, " ", "", StrNumbers),
5 |     maplist(number_string, List, StrNumbers).
6 |
7 | % Проверка, есть ли элемент в списке
8 | member(X, [_|_]).
9 | member(X, [_|Tail]) :-
10 |     member(X, Tail).
11 |
12 | % Пересечение двух списков
13 | intersection([], _, []). % Если первый список пустой, пересечение –
    пустой список
14 | intersection([Head|Tail], List2, [Head|Result]) :-
15 |     member(Head, List2), % Если Head содержится в List2, добавляем его в
    результат
16 |     intersection(Tail, List2, Result).
17 | intersection([_|Tail], List2, Result) :-
18 |     intersection(Tail, List2, Result).
19 |
20 | % Основной предикат для запуска программы
21 | find_intersection :-
22 |     read_list(List1),
23 |     read_list(List2),
24 |     intersection(List1, List2, Intersection),
25 |     Intersection \= [], % Если пересечение непустое
26 |     maplist(write_number_with_space, Intersection),
27 |     nl.
28 |
29 | % Вспомогательный предикат для вывода числа с пробелом после него
30 | write_number_with_space(Number) :-
31 |     write(Number), write(' ').
```

Второе задание:

```
1 | % Чтение строк из файла
2 | read_file_lines(Filename, Lines) :-
3 |     open(Filename, read, Stream),
4 |     read_file_lines_aux(Stream, Lines),
5 |     close(Stream).
6 | read_file_lines_aux(Stream, []) :- at_end_of_stream(Stream), !.
7 | read_file_lines_aux(Stream, [Line | Rest]) :-
8 |     read_line_to_string(Stream, Line),
9 |     read_file_lines_aux(Stream, Rest).
10|
11| % Сравнение двух списков и нахождение позиции первого различия
12| compare_lists([], [], Position, Position) :- !.
13| compare_lists([Char1 | Rest1], [Char1 | Rest2], TempPosition, Position)
14| :-
15|     NextTempPosition is TempPosition + 1,
16|     compare_lists(Rest1, Rest2, NextTempPosition, Position), !.
17| compare_lists(_, _, Position, Position) :- !.
18|
19| % Сравнение двух списков строк
20| compare_files([], [], _) .
21| compare_files([Line1 | Rest1], [Line2 | Rest2], LineNumber) :-
22|     (Line1 \= Line2 ->
23|         % Если строки отличаются, определяем позицию первого различия
24|         string_chars(Line1, List1),
25|         string_chars(Line2, List2),
26|         compare_lists(List1, List2, 1, Position),
27|         format("Файлы различны в строке ~d с символа ~d~n", [LineNumber,
28|             Position])),
29|     false
30|     ; % иначе продолжаем сравнение строк
31|     true
32|     ),
33|     NewLineNumber is LineNumber + 1,
34|     compare_files(Rest1, Rest2, NewLineNumber).
35|
36| compare_text_files(File1, File2) :-
37|     read_file_lines(File1, Lines1),
38|     read_file_lines(File2, Lines2),
39|     compare_files(Lines1, Lines2, 1).
```

РАБОТА ПРОГРАММ

Пример работы первой программы:

```
?- find_intersection.
|: 1 2 3 4 5
|: 2 4 6 7 8
2 4
true .

?- find_intersection.
|: 1 2 3
|: 4 5 6
false.
```

Содержимое файлов для примеров:

false1.pl, true1.pl, true2.pl	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus luctus urna sed urna ultricies ac tempor dui sagittis.
false2.pl	Lorem ipsum dolor sit amet, consectetur Vivamus luctus urna sed urna ultricies ac tempor dui sagittis.

Пример работы второй программы:

```
?- compare_text_files('false1.txt', 'false2.txt').  
Файлы различны в строке 2 с символа 12  
false.  
  
?- compare_text_files('true1.txt', 'true2.txt').  
true.
```