

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

Кафедра прикладной математики и кибернетики

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

Вариант 3

Выполнил:

студент группы ИС-142

Григорьев Ю.В.

ФИО студента

Работу проверил:

\_\_\_\_\_  
ФИО преподавателя

Новосибирск 2023 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].
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.

## РЕШЕНИЕ

### Листинг

#### 1. Файл rgz1.pl

```
% is element member of list or not
member(X, [X | _]).
member(X, [_ | T]) :- member(X, T).

% lists intersection
intersection([], _, []).
intersection([H | T], L2, [H | Result]) :-
    member(H, L2),
    intersection(T, L2, Result).
intersection([_ | T], L2, Result) :-
    intersection(T, L2, Result).

test_intersection :-
    writeln('Введите список 1: '), read(List1), nl,
    writeln('Введите список 2: '), read(List2), nl,
    intersection(List1, List2, Result),
    write('Пересечение списков: '), write(Result), nl.
```

#### 2. Файл rgz2.pl

```
% reading lines from file
read_file_lines(Filename, Lines) :-
    open(Filename, read, Stream),
    read_file_lines_aux(Stream, Lines),
    close(Stream).
```

```

read_file_lines_aux(Stream, []) :- at_end_of_stream(Stream), !.
read_file_lines_aux(Stream, [Line | Rest]) :-
    read_line_to_string(Stream, Line),
    read_file_lines_aux(Stream, Rest).

% comparing two lists and finding first difference position
compare_lists([], [], Position, Position) :- !.
compare_lists([Char1 | Rest1], [Char1 | Rest2], TempPosition, Position) :-
    NextTempPosition is TempPosition + 1,
    compare_lists(Rest1, Rest2, NextTempPosition, Position), !.
compare_lists(_, _, Position, Position) :- !.

% compare two string lists
compare_files([], [], _).
compare_files([Line1 | Rest1], [Line2 | Rest2], LineNumber) :-
    (Line1 \= Line2 =>
        % if strings differ, write first difference
        string_chars(Line1, List1),
        string_chars(Line2, List2),
        compare_lists(List1, List2, 1, Position),
        format("Files differ on line ~d at position ~d~n", [LineNumber, Position])
    ; % else - files are equal
        true
    ),
    NewLineNumber is LineNumber + 1,
    compare_files(Rest1, Rest2, NewLineNumber).


compare_text_files(File1, File2) :-
    read_file_lines(File1, Lines1),
    read_file_lines(File2, Lines2),
    compare_files(Lines1, Lines2, 1).

```

## ДЕМОНСТРАЦИЯ

### Задание 1.

Работа программы:

 test\_intersection.

Введите список 1:

[1,2,3,4,5]

Введите список 2:

[5,2,4]

Пересечение списков: [2, 4, 5]

**true**

## Задание 2.

Файлы equal1.txt и equal2.txt:

```
> cat equal1.txt equal2.txt
welcome
welcome
```

Файлы diff1.txt и diff2.txt:

```
> cat diff1.txt diff2.txt
hello
dear
brother

hello
dea
brother
```

## Работа программы:

```
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- consult('/Users/allenvox/rgz2.pl').
true.

?- compare_text_files('/Users/allenvox/equal1.txt', '/Users/allenvox/equal2.txt').
true.

?- compare_text_files('/Users/allenvox/diff1.txt', '/Users/allenvox/diff2.txt').
Files differ on line 2 at position 4
true.

?-
```