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

Кафедра прикладной математики и кибернетики

Лабораторная работа № 5
по дисциплине «Функциональное и логическое программирование»

Бригада №2

Выполнили:
студент группы ИВ-823
Лозовой Владислав Александрович
ФИО студента

студент группы ИВ-823
Шиндель Эдуард Дмитриевич
ФИО студента

Работу проверил:
ассистент кафедры Агалаков А.А.
ФИО преподавателя

Новосибирск 2020 г.

ОГЛАВЛЕНИЕ

ЗАДАНИЕ

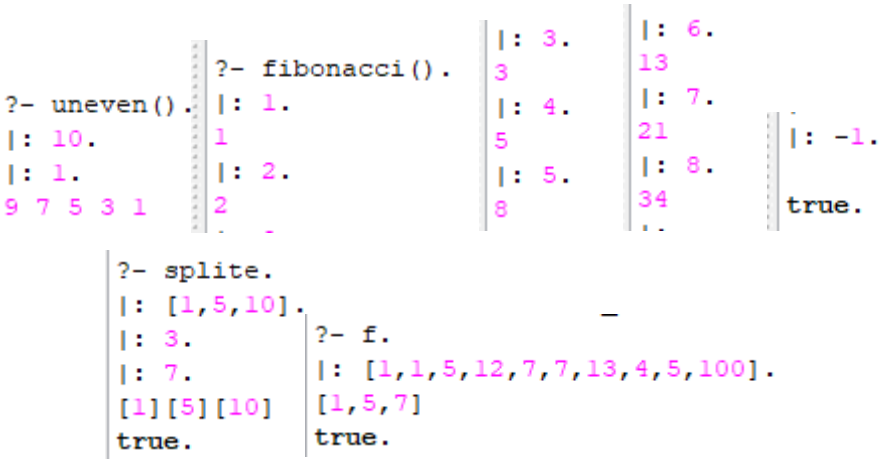
1. Написать предикат, который печатает все нечётные числа из диапазона в порядке убывания. Границы диапазона вводятся с клавиатуры в процессе работы предиката.

2. Написать предикат, который находит числа Фибоначчи по их номерам, которые в цикле вводятся с клавиатуры. Запрос номера и нахождение соответствующего числа Фибоначчи должно осуществляться до тех пор, пока не будет введено отрицательное число. Циклический ввод организовать с помощью предиката repeat. Числа Фибоначчи определяются по следующим формулам: F(0)=1, F(1)=1, F(i)=F(i-2)+F(i-1) (i=2, 3, 4, ...).

3. Написать предикат, который разбивает числовой список по двум числам, вводимым с клавиатуры на три списка: меньше меньшего введенного числа, от меньшего введенного числа до большего введенного числа, больше большего введенного числа. Список и два числа вводятся с клавиатуры в процессе работы предиката. Например: [3,7,1,-3,5,8,0,9,2], 8, 3 –> [1,-3,0,2], [3,7,5,8], [9].

4. Написать предикат, который формирует список из наиболее часто встречающихся элементов списка. Список вводится с клавиатуры в процессе работы предиката. Встроенные предикаты поиска максимума и сортировки не использовать! Например: [0,3,5,7,1,5,3,0,3,3,5,7,0,5,0] –> [0,3,5].

ДЕМОНСТРАЦИЯ РАБОТЫ ПРОГРАММЫ



ВЫВОД

Было выполнено 4 задачи в Prolog.

Написаны предикат для вывода нечетных чисел из диапазона, предикат для вычисления числа Фибоначчи по их номеру, предикат для разбиения списка и предикат для формирования списка из часто встречающихся чисел.

ПРИЛОЖЕНИЕ

```
/* №1 Написать предикат, который печатает все нечётные числа из
диапазона в
 порядке убывания. Границы диапазона вводятся с клавиатуры в
процессе работы предиката. */

even(X) :- X mod 2 =:= 0.

print_uneven(X, Y) :- not(even(Y)), write(Y), tab(1), fail.

print_uneven(X, Y) :- Y \== X, Y1 is Y - 1, print_uneven(X, Y1).

uneven() :- read(X), read(Y), print_uneven(Y, X).

/* №2 Написать предикат, который находит числа Фибоначчи по их
номерам, которые в цикле
вводятся с клавиатуры. Запрос номера и нахождение соответствующего
числа
Фибоначчи должно осуществляться до тех пор, пока не будет введено
отрицательное
число. */

calc_fibonacci(0, Y) :- Y is 1, !.
calc_fibonacci(1, Y) :- Y is 1, !.

calc_fibonacci(X, Y) :-
        X1 is X - 2,
        X2 is X - 1,
        calc_fibonacci(X1, Y1),
        calc_fibonacci(X2, Y2),
        Y is Y1 + Y2.

fibonacci() :-
        repeat,
        read(X),
        (X < 0, !; calc_fibonacci(X, Y), write(Y), nl, fail).

/* №3 Написать предикат, который разбивает числовой список по двум
числам, вводимым
с клавиатуры на три списка: меньше введенного числа, от меньшего
введенного
числа до большего введенного числа, больше большего введенного
числа.
```

```
Список и два числа вводятся с клавиатуры в процессе работы
предиката. */

splite :- read(List),read(A),read(B),splite_prepare(A, B, Min,
Max),
          splite_rec(List, Min, Max, List1, List2, List3),
          write(List1), write(List2), writeln(List3).

        splite_prepare(A, B, Min, Max) :- A > B, !, Max = A, Min =
B; Max = B, Min = A.

        splite_rec([], _A, _B, [], [], []).

        splite_rec([Head|Tail], A, B, [Head|List1], List2, List3)
:-
                Head < A, !,splite_rec(Tail, A, B, List1, List2,
List3).
        splite_rec([Head|Tail], A, B, List1, [Head|List2], List3)
:-
                Head =< B, !, splite_rec(Tail, A, B, List1, List2,
List3).
        splite_rec([Head|Tail], A, B, List1, List2, [Head|List3])
:-
                !, splite_rec(Tail, A, B, List1, List2, List3).

/* №4 Написать предикат, который формирует список из наиболее часто
встречающихся
элементов списка. Список вводится с клавиатуры в процессе работы
предиката. */

foo :- read(List), msort(List, SList), cou(SList, _, ResList),
write(ResList).

        cou([], 0, []).
        cou([H|B], K, [H]) :-
                foo([H|B], H, K, List), cou(List, MX, _), K > MX,
!.
        cou([H|B], MX, List1) :-
                foo([H|B], H, K, List), cou(List, MX, List1), K <
MX, !.
        cou([H|B], MX, [H|List1]) :-
                foo([H|B], H, K, List), cou(List, MX, List1), K =:=
MX, !.
```

```
        foo([], _, 0, []) :- !.
        foo([H|B], X, 0, [H|B]) :- H =\= X, !.
        foo([H|B], X, K, B1) :-  H =:= X,  foo(B, X, K1, B1), K is
K1 + 1.
```