

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 №5

Выполнили:

студенты группы ИП-811

Разумов Д.Б.

Макаров Д.А.

Работу проверил: Галкина М.Ю.

Новосибирск 2020 г.

## Оглавление

Задание.....	3
Листинг программы.....	4
Скриншоты.....	8

## 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:  $F(0)=1$ ,  $F(1)=1$ ,  $F(i)=F(i-2)+F(i-1)$  ( $i=2, 3, 4, \dots$ ).
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  $\rightarrow [1, -3, 0, 2]$ ,  $[3, 7, 5, 8]$ ,  $[9]$ .
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:  $[0, 3, 5, 7, 1, 5, 3, 0, 3, 3, 5, 7, 0, 5, 0]$   $\rightarrow [0, 3, 5]$ .

# Листинг программы

```
% ЗАДАНИЕ 1

task1(A, B):-
    A > B, !.

task1(_, B):-
    M is B mod 2, M == 1, writeln(B), fail.

task1(A, B):-
    B1 is B - 1, task1(A, B1).

% проверка:
% task1(10, 20).
% task1(-6, 5).

% ЗАДАНИЕ 2

fibonacci(Index, Value):-
    Index =< 1, Value is 1, !;
    I1 is Index - 1, I2 is Index - 2,
    fibonacci(I1, V1), fibonacci(I2, V2),
    Value is V1 + V2.

task2:-
    repeat,
    writeln('enter index of number Fibonacci '), read(Index),
    (Index < 0, !;
    fibonacci(Index, Value), writeln(Value), fail).

% проверка:
% task2.
```

```

% ЗАДАНИЕ 3

task3(A, B, [H1|T1], [H1|T2], L2, L3):-
    H1 @< A, !, task3(A, B, T1, T2, L2, L3).
task3(A, B, [H1|T1], L1, [H1|T2], L3):-
    H1 @>= A, H1 @=< B, !, task3(A, B, T1, L1, T2, L3).
task3(A, B, [H1|T1], L1, L2, [H1|T2]):-
    task3(A, B, T1, L1, L2, T2).
task3(_, _, [], [], [], []).

% проверка:
% task3(3, 8, [3,7,1,-3,5,8,0,9,2], L1, L2, L3).
% task3(-2, 5, [5,4,5,6,-12,12,-3,0,21,1], L1, L2, L3).
% task3(0, 11, [0, 11, -11], L1, L2, L3).

% ЗАДАНИЕ 4

% деление списка на две части по числу M:
split(M, [H1|T1], [H1|T2], L2):-
    H1 @< M, !, split(M, T1, T2, L2).
split(M, [H1|T1], L1, [H1|T2]):-
    split(M, T1, L1, T2).
split(_, [], [], []).

% быстрая сортировка:
qsort([], []).
qsort([Elem], [Elem]).
qsort([Head|Tail], Sort_list):-
    split(Head, Tail, Less, More),
    qsort(More, Sort_more),
    qsort(Less, Sort_less),!,
    append(Sort_less, [Head|Sort_more], Sort_list).

```

```

% узнать сколько раз (частота, frequency) встречается число:
set_frequency(Input_list, Output_list):-
    set_frequency(Input_list, [], Temp1_list),
    qsort(Temp1_list, Temp2_list),
    reverse(Temp2_list, Output_list).

% точка возврата:
set_frequency([], List, List).

% если Num встречено не впервые:
set_frequency([Num | Tail], Temp1_list, Return_list):-
    select([Freq, Num], Temp1_list, Temp2_list),
    Freq1 is Freq + 1, !,
    set_frequency(Tail, [[Freq1, Num] | Temp2_list], Return_list).

% если Num встречено впервые:
set_frequency([Num | Tail], Temp0_list, Return_list):-
    set_frequency(Tail, [[1, Num] | Temp0_list], Return_list).

% записать в список наиболее встречаемые числа:
max_frequency(Max, [[Freq, Num]|T1], [Num|T2]):-
    Max == Freq, max_frequency(Max, T1, T2).
max_frequency(Max, [[_, _]|T1], Max_list):-
    max_frequency(Max, T1, Max_list).
max_frequency(_, [], []):- !.

% запустить все это:
task4(Input_list, Max_list):-
    set_frequency(Input_list, [[Max_freq, Num]|Tail]),
    max_frequency(Max_freq, [[Max_freq, Num]|Tail], Max_list), !.

```

```
% проверка:  
% set_frequency([0,3,5,7,1,5,3,0,3,3,5,7,0,5,0], L).  
% task4([0,3,5,7,1,5,3,0,3,3,5,7,0,5,0], L).  
% task4([2,4,1,4,4,4,2], L).  
% task4([-1, 1], L).
```

# Скриншоты

```
?- task1(10, 20).  
19  
17  
15  
13  
11  
true.
```

```
?- task1(-6, 5).  
5  
3  
1  
-1  
-3  
-5  
true.
```

Выполнение задания 1.

```
?- task2.  
enter index of number Fibonacci  
| 0.  
1  
enter index of number Fibonacci  
| 2.  
2  
enter index of number Fibonacci  
| 4.  
5  
enter index of number Fibonacci  
| 6.  
13  
enter index of number Fibonacci  
| -2.  
  
true.
```

Выполнение задания 2.



?- task3(3, 8, [3,7,1,-3,5,8,0,9,2], L1, L2, L3).  
L1 = [1, -3, 0, 2],  
L2 = [3, 7, 5, 8],  
L3 = [9] .

?- task3(0, 11, [0, 11, -11], L1, L2, L3).  
L1 = [-11],  
L2 = [0, 11],  
L3 = [].

Выполнение задания 3.

?- task4([0,3,5,7,1,5,3,0,3,3,5,7,0,5,0], L).  
L = [5, 3, 0].

?- task4([-1, 1], L).  
L = [1, -1].

Выполнение задания 4.