

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 № 4

Выполнили:

студенты группы ИП-811

Бобова И.А
ФИО студента

Адов А. С.
ФИО студента

Работу проверил: Галкина Марина
Юрьевна

ФИО преподавателя

Новосибирск 2020 г.

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]`.

Код программы и визуализация

% ЗАДАНИЕ 1

oddNumbers(Left, Right):-

Left > Right, !.

oddNumbers(_, Right):-

Remainder is Right mod 2, Remainder == 1, writeln(Right), fail.

oddNumbers(Left, Right):-

Temp is Right - 1, oddNumbers(Left, Temp).

% проверка:

% oddNumbers(10, 20).

% oddNumbers(-6, 5).

% ЗАДАНИЕ 2

valueSearch(I, F):-

I =< 1, F is 1, !;

I1 is I - 1, I2 is I - 2,

valueSearch(I1, F1), valueSearch(I2, F2),

F is F1 + F2.

fibonacciNumbers:-

repeat,

writeln('Enter index of number Fibonacci: '), read(I),

(I < 0, !;

valueSearch(I, F), writeln(F), fail).

% проверка:

% fibonacciNumbers.

% ЗАДАНИЕ 3

splittingList(A, B, [H1|T1], [H1|T2], L2, L3):-

H1 @< A, !, splittingList(A, B, T1, T2, L2, L3).

splittingList(A, B, [H1|T1], L1, [H1|T2], L3):-

H1 @>= A, H1 @=< B, !, splittingList(A, B, T1, L1, T2, L3).

splittingList(A, B, [H1|T1], L1, L2, [H1|T2]):-

splittingList(A, B, T1, L1, L2, T2).

splittingList(_, _, [], [], [], []).

% проверка:

% splittingList(3, 8, [3,7,1,-3,5,8,0,9,2], 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([2,4,1,4,4,2], L).
```

```
% task4([-1, 1], L).
```

Результаты работы

Задание 1

```
?- oddNumbers(1, 10).
```

```
9
7
5
3
1
```

```
true.
```

```
?- |
```

```
?- oddNumbers(-10, 3).
```

```
3
1
-1
-3
-5
-7
-9
```

```
true.
```

```
?- |
```

Задание 2

```
?- fibonacciNumbers.
```

```
Enter index of number Fibonacci:
```

```
|: 4.
```

```
5
```

```
Enter index of number Fibonacci:
```

```
|: 8.
```

```
34
```

```
Enter index of number Fibonacci:
```

```
|: -2.
```

```
true.
```

```
?- |
```

Задание 3

```
?- splittingList(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] ;
```

```
false.
```

```
?- |
```

Задание 4

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

```
L = [5, 3, 0].
```

```
?- task4([5, 8, 8, 8, 8, 1, 2], L).
```

```
L = [8].
```

```
?- task4([-9, 1], L).
```

```
L = [1, -9].
```

```
?- |
```