МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

## Факультет информационных технологий и робототехники (ФИТР)

Кафедра программного обеспечения информационных систем

и технологий

**Отчет по лабораторной работе № 6**

по дисциплине: ”Функциональное и логическое программирование”

# на тему: “Списки.”

Выполнил**:** студент гр. 10702118

Дубоделов А.В.

Принял**:** Ковалькова И.А.

Минск 2019

**Цель лабораторной работы:** приобретение  практических  навыков составления и отладки программ с использованием Отсечения.

**Задание:**

3.        **Содержание задания по лабораторной работе**.

Из  списка  L1  получить список L2, очередной элемент которого равен среднему  арифметическому  очередной  тройки элементов  списка  L1.  Если  число  элементов списка L1 не кратно 3, то последний элемент списка L2 получается делением на 3 одного или суммы двух  последних  элементов  списка L1.  Список L1 вводится по подсказке с экрана. В результате выполнения программы должны выдаваться исходный  L1  и  результирующий L2 списки.

domains

listr = real\*

predicates

nondeterm result

nondeterm solution(listr,listr) %(i,0)

nondeterm sum(listr,listr,real,real,integer) %(i,0,0,i,i)

clauses

solution([],[]). %если список 0-й

solution(List,NewList) :- sum(List,ReturnList,ReturnSum,0.0,1), solution(ReturnList,NewList1), NewList=[ReturnSum|NewList1]. %если список не 0-й

sum([Head|Tail],ReturnList,ReturnSum,Sum,Amount) :- Amount<4, Sum1=Sum+Head/3, Amount1=Amount+1, sum(Tail,ReturnList,ReturnSum,Sum1,Amount1). %будет происходить пока не просчитает 3 числв

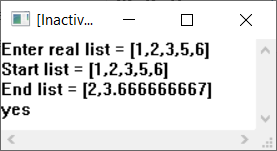
sum(A,A,Sum,Sum,Amount) :- Amount>3. %если 3 раза прошло, вернет A и Sum как ReturnList и ReturnSum

sum([],[],Sum,Sum,\_). %если были переданы пустые списки

result :- write("Enter real list = "),readterm(listr,List),solution(List,List1),write("Start list = ",List),nl,write("End list = ",List1),nl.

goal

result.



4.5 Удалить все вхождения введенного элемента

domains

list = integer\*

predicates

delete(list ,integer, list)

result

clauses

delete([],\_,[]).

delete([H|T],El,[H|T1]):- H<>El,!,delete(T,El,T1).

delete([\_|T],El,T1):- delete(T,El,T1).

result:-write("Enter list: "), readterm(list,L),

write("Enter integer items to be deleted: "),nl,

write("I="),readint(I),

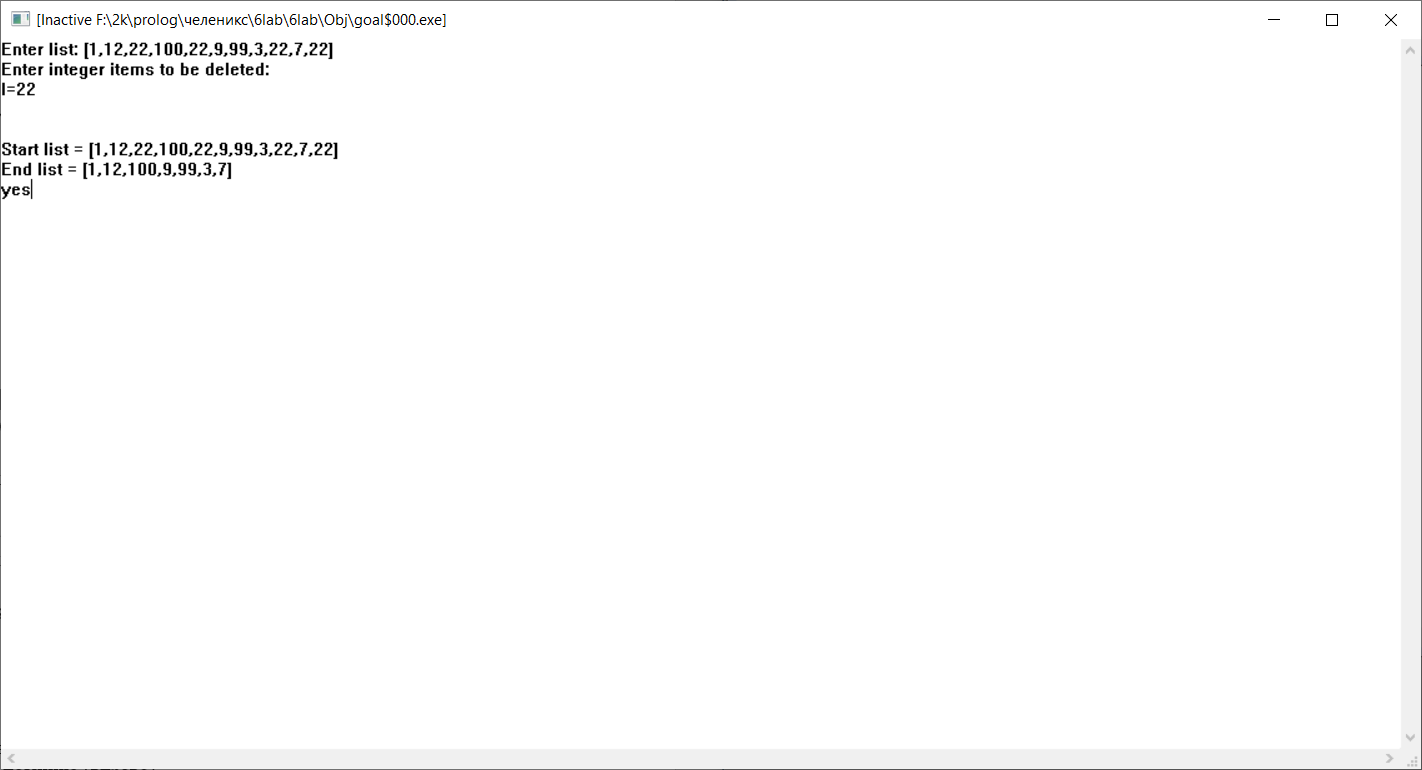
delete(L,I,K),nl,nl,

write("Start list = ",L),nl,

write("End list = ",K),nl.

goal

result.



4.17 Уменьшить каждый элемент списка на среднее арифметическое всех членов списка

domains

listr = real\*

predicates

sum(listr, real)

len(listr, integer)

enc\_list(listr, listr, real)

result2

clauses

sum([],0.0). %сумма нулевого списка 0

sum([H|T],S) :- sum(T,S1), S = S1 + H. %если список не 0-й

len([],0). %длина нулевого списка 0

len([\_|T],N):- len(T,N1), N=N1+1. %если списко не 0-й

enc\_list([],[],\_).

enc\_list([H1|T1],[H2|T2], A):- H2=H1+A,enc\_list(T1,T2,A).

result2 :- write("Enter real list: "), readterm(listr,L),nl,

sum(L,S),write("Sum of items = ",S),nl,

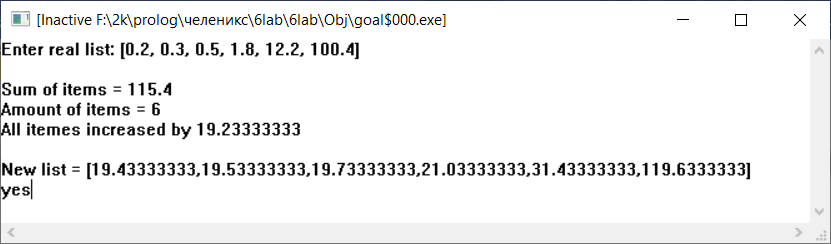
len(L,N),write("Amount of items = ", N),nl,

A = S/N,write("All itemes increased by ",A),nl,nl,

enc\_list(L,L1,A) ,write("New list = ", L1),nl.

goal

result2.



Контрольные вопросы:

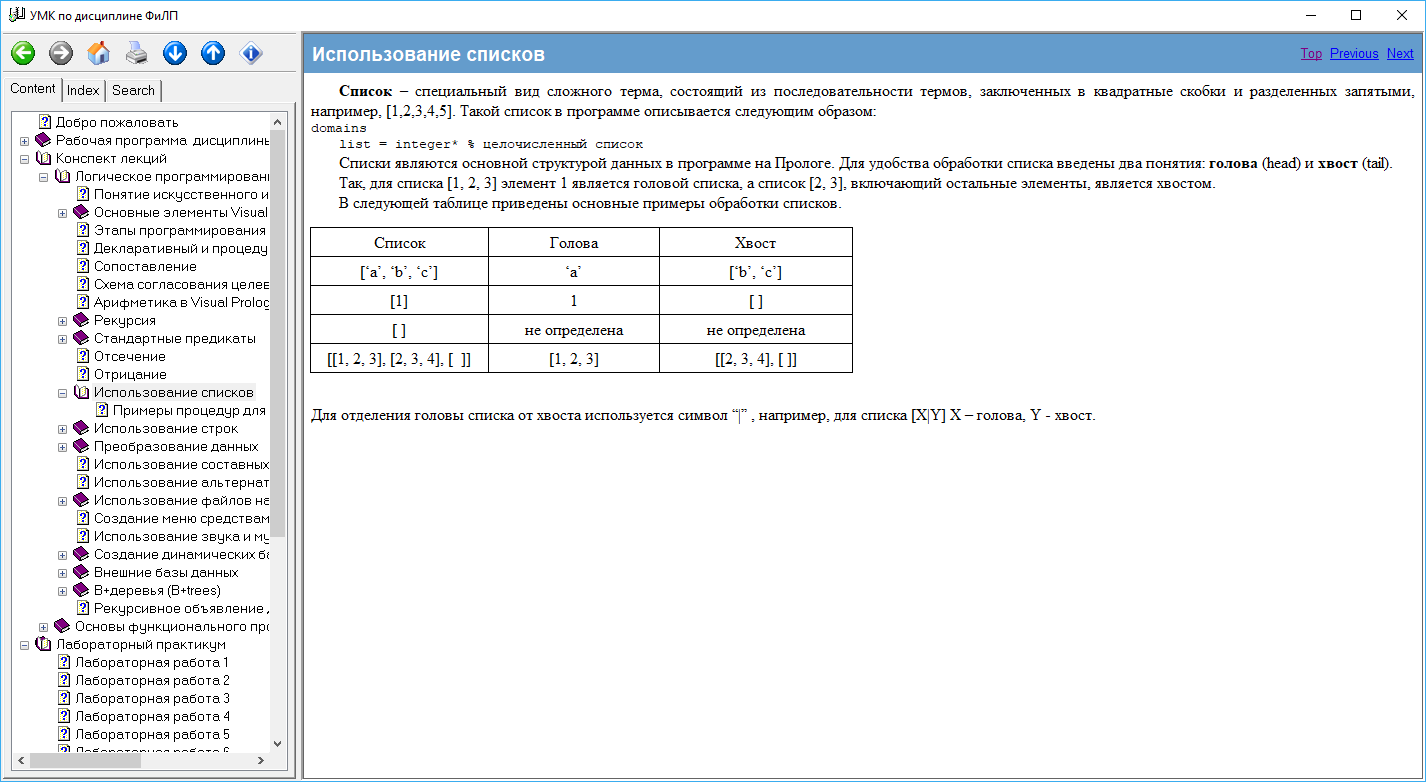
**1.**

Что такое список ?

**Список** – специальный вид сложного терма, состоящий из последовательности термов, заключенных в квадратные скобки и разделенных запятыми, например, [1,2,3,4,5]

**2.**

Голова и хвост списка. Изображение списка в виде головы и хвоста.



**3.**

Как обозначается пустой список ?

[ ]

**4.**

Можно ли изобразить пустой список в виде головы и хвоста ?

Нет.

**5.**

Что означают следующие виды изображения списка: L, [X], [ \_ , A], [ \_ |T], [H| \_ ], [ \_ ], [ \_ ,B|C] ?

L-тип списка

[X] -список состоящий из 1 элемента

[\_,A] –  список из двух элементов, первый элемент может принимать любое значение.

[\_|T] – список в котором голова принимает любое значение

[H|\_] – список в котором не важен хвост

[\_] – список из одного элемента значение которого не имеет значения

[\_,B|C] – список в голове которого 2 элемента, и первый может принимать любое значение.

**6.**

Может ли быть в голове списка более одного элемента ?

Нет, только если другой список

**7.**

Принадлежит ли элемент 5 списку [[1, 2, 3], [4, 5, 6]] ?

Нет

**8.**

Сопоставимы ли два списка: [5, 6, 7] и [H|T] ?

Да H=5 T=[6,7]

**9.**

Можно ли в процедурах ввода списка ввод\_списка1 и ввод\_списка2 поменять местами первое и второе предложения ?

Нет.

**10.**

Как нужно изменить процедуру удал\_перв\_эл, чтобы можно было удалить из списка: a) два первых элемента; б) второй элемент ?

А) удал\_перв\_эл([ \_ ,\_| T], T).

B) удал\_перв\_эл([ H,\_ | T], [H|T]).

**11.**

Как нужно изменить процедуру удал\_посл\_эл, чтобы из списка были удалены: a) два последние элементы; б) предпоследний элемент ?

А)

удал([\_,\_],[]).

удал([H|T],[H|T1]) if удал(T,T1).

B)

удал [\_,X],[X]).

удал([H|T],[H|T1]) if удал(T,T1).

**12.**

Как нужно изменить процедуру выдел\_перв\_эл, чтобы можно было выделить из списка: a) два первых элемента; б) второй элемент ?

А)

выдел\_перв\_эл([H,H1| \_ ], H,H1).

B)

выдел\_перв\_эл([\_,H1| \_ ], H1).

**13.**

Как нужно изменить процедуру выдел\_посл\_эл, чтобы из списка можно было выделить: a) два последние элементы; б) предпоследний элемент ?

А)

выдел\_посл\_эл([X,H], X,H).

выдел\_посл\_эл([ \_ |T], X,H):- выдел\_посл\_эл(T, X,H).

B)

выдел\_посл\_эл([X,\_], X).

выдел\_посл\_эл([ \_ |T], X):- выдел\_посл\_эл(T, X).

**14.**

Как нужно изменить процедуру доб\_эл\_в\_начало, чтобы можно было вставить заданный элемент между первым и вторым элементами списка ?

доб\_эл\_в\_начало(E, [H,H1|T], [H,E,H1|T]).

**15.**

Как нужно изменить процедуру доб\_эл\_в\_конец, чтобы заданный элемент вставился перед последним элементом списка ?

Изменить граничное условие таким образом, что бы он останавливался на последнем элементе.

доб\_эл\_в\_конец(E, [H], [E,H]).

доб\_эл\_в\_конец(E, [H|T], [H|T1]):- доб\_эл\_в\_конец(E, T, T1).

**16.**

Можно ли в программе со списками обойтись без раздела domains ?.

нет.