МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

## Факультет информационных технологий и робототехники (ФИТР)

Кафедра программного обеспечения вычислительной техники

и автоматизированных систем

**Отчет по лабораторной работе № 6**

по дисциплине: ”Функциональное и логическое программирование”

# на тему: ”Работа со списками”

Выполнили**:** студенты гр. 10701118

Воробей И.А.

Принял**:** Ковалькова И.А.

Минск 2019

**Цель лабораторной работы:**  приобретение навыков работы со  списками в программах на Visual Prolog.

**Задание1:** 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.

**Код программы:**

domains

list = integer\*

list2 = real\*

predicates

nondeterm sum(list,list2)

%sum(x,y)(integer\*,real\*):(i,o)

nondeterm read\_list(list)

%read\_list(x)(integer\*):(i)

clauses

read\_list([H|T]) :- readint(H),!,read\_list(T).

read\_list([]).

sum([],[]).

sum([A,B,C|T],[Sum|T1]):-Sum = (A+B+C)/3,sum(T,T1).

sum([A, B], [Sum]) :- Sum = (A+B)/3.

sum([A], [Sum]) :- Sum = A/3.

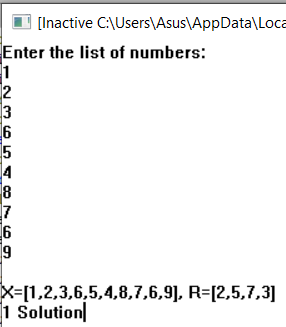
goal

write("Enter the list of numbers: "),nl,

read\_list(X),

sum(X,R).

**Результаты:**



**Задание 4.9 а:**Произвести циклический сдвиг элементов списка:

                    - на одну позицию влево (вправо).

**Код программы:**

domains

list=real\*

predicates

nondeterm read\_list(integer,list)

%read\_list(x,r)(integer,real\*):(i,o)

sdvig(list, list)

%sdvig(x,r)(real\*,real\*):(i,o)

conc(list, list, list)

%conc(x,r)(real\*,real\*,real\*):(i,i,o)

nondeterm result

clauses

read\_list(0,[]).

read\_list(N,[H|T]):-readreal(H),N1=N-1,read\_list(N1,T).

conc([],List,List).

conc([X|L1],List2,[X|L3]):-conc(L1,List2,L3).

sdvig( [X|L1],List2) :-conc(L1,[X], List2).

result:-write("Enter number of elements"),

write("N="),readint(N),

write("Enter instances of list:"),nl,

read\_list(N,L),

write("Vergin list L="),

write(L),nl,

write("New list L1="),

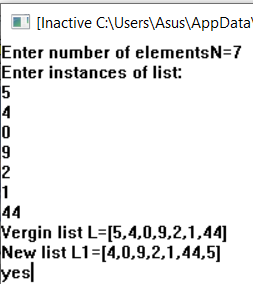
sdvig( L ,L1),

write(L1),nl.

goal

result.

**Результаты:**



**Результаты:**

**Задание4.14:** Вывести первые N положительных элементов списка.

**Код программы:**

domains

list=real\*

predicates

nondeterm read\_list(integer,list)

%read\_list(x,r)(integer,real\*):(i,o)

positive\_elements(list,integer,list)

%positive\_elements(real\*,integer,real\*):(i,i,o)

nondeterm result

clauses

read\_list(0,[]).

read\_list(N,[H|T]):-readreal(H),N1=N-1,read\_list(N1,T).

positive\_elements(\_,0,[]):-!.

positive\_elements([Head|Tail],N1,[Head|TailPositive]):-

Head > 0,N=N1-1,positive\_elements(Tail,N, TailPositive),!.

positive\_elements([\_Head|Tail],N1, TailPositive):-

positive\_elements(Tail,N1, TailPositive).

result:-write("Enter number of elements"),

write("N="),readint(N),

write("Enter instances of list:"),nl,

read\_list(N,L),

write("Enter number of first positev elements"),

write("N1="),readint(N1),

write("Vergin list L="),

write(L),nl,

write("Positive list L1="),

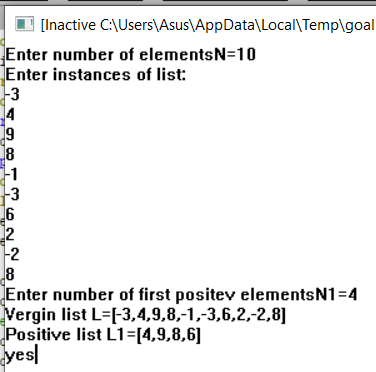
positive\_elements(L,N1,L1),

write(L1),nl.

goal

result.

**Результаты:**



**Контрольные вопросы:**

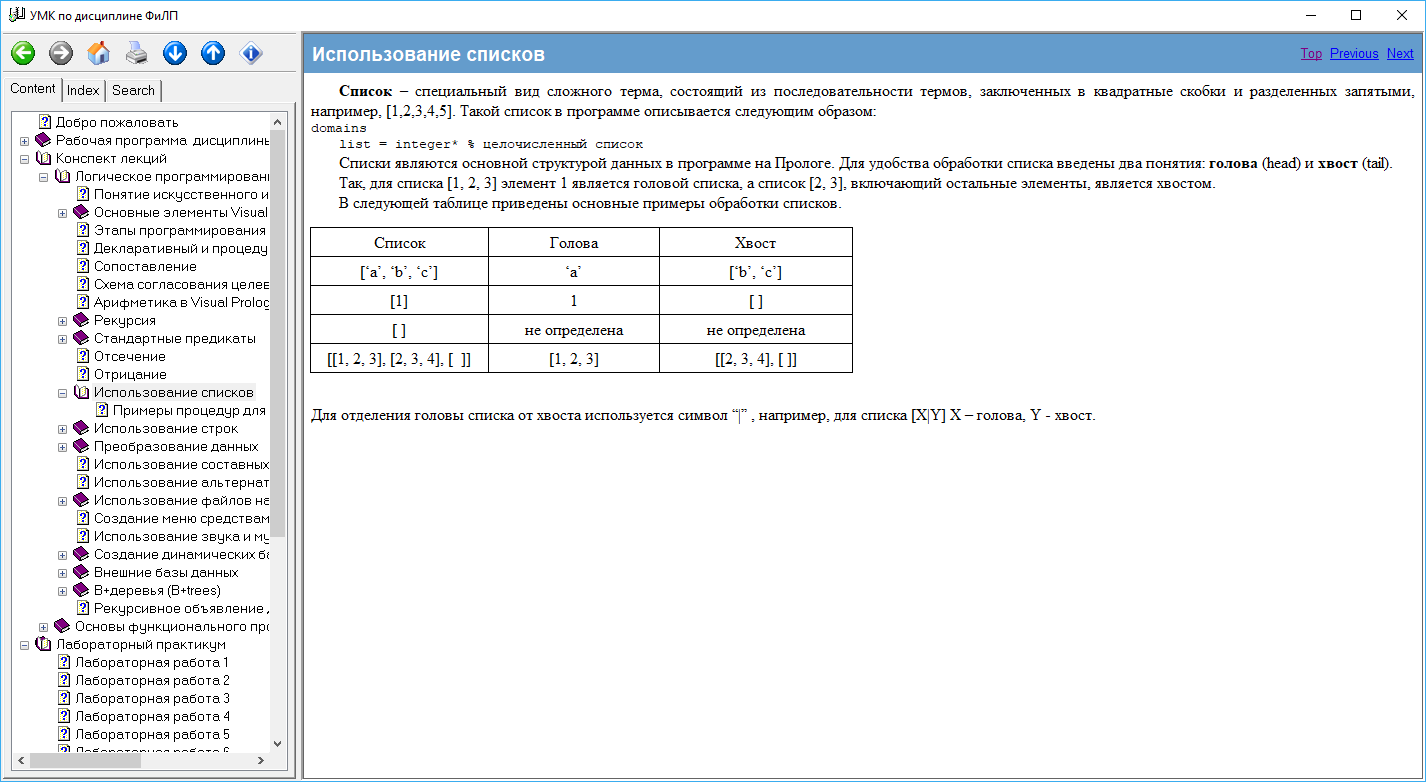
**1.**

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([H,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.

**Выводы:** *приобрели практические навыки использования списков в программах в Пролог-программах.*