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

(СибГУТИ)

Кафедра прикладной математики и кибернетики

РАСЧЕТНО-ГРАФИЧЕСКОЕ ЗАДАНИЕ

по дисциплине «Функциональное и логическое программирование»

Вариант 11

Выполнил:

студент гр. И\*–\*\*\*

Некто Н.Н.

Проверил:Ассистент Кафедры ПМиК

Бочкарев Б.В.

Новосибирск 2019 г.

**Задание**

1. Список целых чисел разделите на два списка: из чётных элементов и нечётных элементов;

Например, [8, 9, 2, -1, 6, -9] → [8, 2, 6], [9, -1, -9].

2. Найдите в файле все слова максимальной длины. Сформируйте новый файл из найденных слов.

**Выполнение**

Задание 1:

Предикат *odd\_even\_sort* распределяет числа по спискам в зависимости от чётности чисел и выводит списки чётных и нечётных чисел на экран. Если число чётное (при делении числа на 2 остаток равен нулю), то оно заносится в список *Even*, иначе – в список *Odd*.

%Задание 1.

odd\_even([], [], []).

odd\_even([Head | Tail], [Head | Even], Odd):-

Head mod 2 =:= 0, !, odd\_even(Tail, Even, Odd).

odd\_even([Head | Tail], Even, [Head | Odd]):-

Head mod 2 =:= 1, !, odd\_even(Tail, Even, Odd).

odd\_even\_sort(List):-

odd\_even(List, Even, Odd),

write('Even: '), write(Even), nl,

write('Odd: '), write(Odd), nl.

%Пример запроса

%?-odd\_even\_sort([8, 7, 9, 2, -1, 6, -9]).

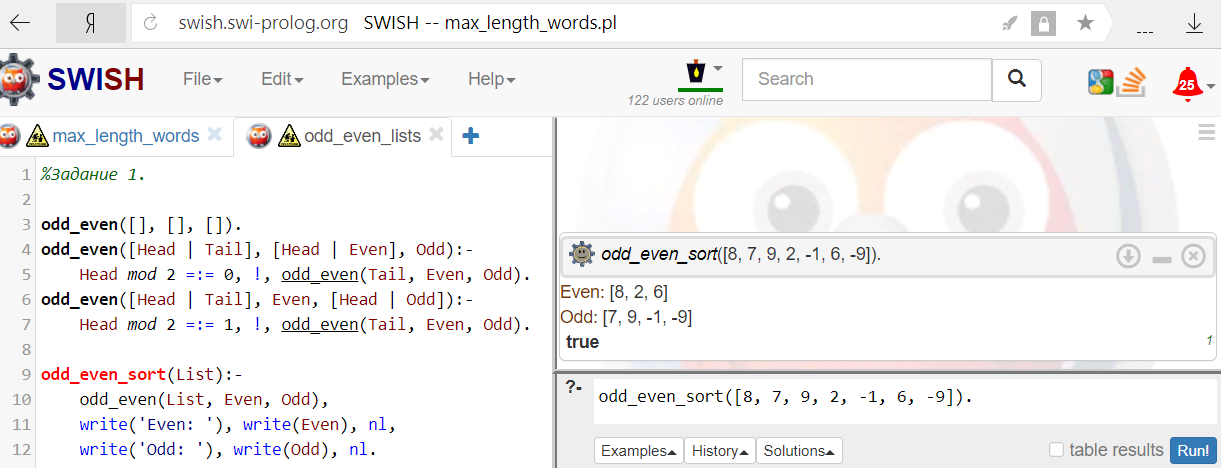


Рисунок 1 – Пример работы 1-ого задания

Задание 2:

1) На вход подаётся строка, состоящая из слов, разделённых пробелом, и заканчивающаяся точкой.

2) В предикат str\_to\_list\_words(S, Lw) на вход подаётся строка(S) и возвращается список(Lw), атомы которого – слова из строки S.

3) В предикат arr\_len\_word(Lw, Ll) подаётся на вход список из слов(Lw), и возвращается список длин этих слов(Ll).

4) Предикат maximum(Ll, MaxLen) в списке длин слов(Ll) находит максимальную длину(MaxLen) слова(ов), которое(ые) есть в списке(Ll).

5) Предикат list\_of\_max\_words(Lw, MaxLen, Ln) из списка слов(Lw) формирует новый список слов(Ln), длина которых равна атому MaxLen.

%Задание 2.

%Найдите в файле все слова максимальной длины.

%Сформируйте новый файл из найденных слов.

%Предикат преобразования заданной строки в список слов

str\_to\_list\_words(S, [Hw | Tn]):-

front\_token(S, W, S2), not(S2 = " "), !,

Hw = W, str\_to\_list\_words(S2, Tn).

str\_to\_list\_words(\_, []).

%Предикат, который выделяет слова из строки.

front\_token(S, W, L1):-

atom\_chars(S, L), %строка > список символов

append(ListW, [' ' | List1], L), !,

atom\_chars(W, ListW),

atom\_chars(L1, List1).

front\_token(S, W, L1):-

atom\_chars(S, L), %строка > список символов

append(ListW, ['.' | List1], L), !,

atom\_chars(W, ListW),

atom\_chars(L1, List1).

%Предикат формирования списка длин слов по списку этих слов

arr\_len\_word([], []):- !.

arr\_len\_word([H | T], [Hn | Tn]):-

atom\_length(H, LenW),

Hn = LenW,

arr\_len\_word(T, Tn).

%Предикат нахождения максимума в списке (длин заданных слов)

maximum([X], X).

maximum([H | T], H):-maximum(T, M), H > M, !.

maximum([\_ | T], M):-maximum(T, M).

%Предикат формирования нового списка путем удаления из заданного списка слов длины, отличной от заданной величины

list\_of\_max\_words([], \_, []):- !.

list\_of\_max\_words([H | T], X, [Hn | Tn]):-

atom\_length(H, LenW), %длина строки

LenW = X, !,

Hn = H,

list\_of\_max\_words(T, X, Tn).

list\_of\_max\_words([\_ | T], X, Ln):- list\_of\_max\_words(T, X, Ln).

%Предикат для записи строки в файл

read\_from\_file(Filename, S):-

open(Filename, read, Input),

readln(Input, S),

close(Input).

%Предикат для записи строки в файл

write\_to\_file(Filename, S):-

open(Filename, write, Out),

write(Out, S),

close(Out).

%предикат для запуска программы

%Input('c:/input\_file.txt').

%Output('c:/output\_file.txt').

%max\_len\_words(Input, Output):-

max\_len\_words(S):-

write("Список слов:"), nl,

%read\_from\_file(Input, S),

str\_to\_list\_words(S, Lw), %формирование списка Lw из слов строки S

write(Lw), nl,

arr\_len\_word(Lw, Ll), %нахождение длин слов и занесение их в список

maximum(Ll, MaxLen), %нахождение максимальной из длин слов

write("Длина наиболее длинного слова: "), write(MaxLen), nl,

list\_of\_max\_words(Lw, MaxLen, Ln), %формирования списка слов макс.длины

%atomics\_to\_string(Ln, " ", Sn), %список > строка

%write\_to\_file('Output, Sn), nl, %запись строки самых длинных слов в файл

write(Ln).

%Строка для запуска

%?-max\_len\_words("red orange yellow green blue violet indigo.").

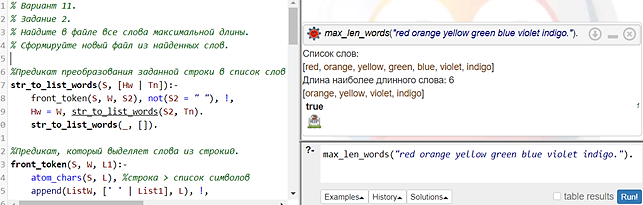


Рисунок 2 – Пример работы 2-ого задания