МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»

Кафедра прикладной математики и кибернетики

РАСЧЕТНО-ГРАФИЧЕСКОЕ ЗАДАНИЕ

по дисциплине «Функциональное и логическое программирование»

Вариант №15

Выполнил:

Ст. гр. ИВ-921

Черемисин И.И.

Проверил:

Ассистент каф. ПМиК

Белевцова Е.А.

Содержание

[Постановка задачи 3](#__RefHeading___Toc5302_1932663523)

[1. Списки (ввод с клавиатуры, использование рекурсии) 3](#__RefHeading___Toc5579_461734499)

[2. Строки, файлы 3](#__RefHeading___Toc5579_4617344991)

3. Динамические базы данных 3

[Ход работы 3](#__RefHeading___Toc5303_3596095987)

[Используемые функции 3](#__RefHeading___Toc5305_3596095987)

[Отдельные функции для 1.19 3](#__RefHeading___Toc5307_3596095987)

[Отдельные функции для 2.19 3](#__RefHeading___Toc5309_3596095987)

Отдельные функции для 3.19 3

[Использованные встроенные функции 3](#__RefHeading___Toc5311_3596095987)

[Выполнение программы 4](#__RefHeading___Toc5345_3596095987)

[Работа программы 1.19 4](#__RefHeading___Toc5347_3596095987)

[Работа программы 2.19 4](#__RefHeading___Toc310_3918689516)

[Работа программы 3.19 5](#__RefHeading___Toc310_3918689516)

[Список литературы 6](#__RefHeading___Toc570_3918689516)

[Листинг 6](#__RefHeading___Toc20158_1932663523)

[Задание 1.19 7](#__RefHeading___Toc2851_3918689516)

[Задание 2.19 7](#__RefHeading___Toc2853_3918689516)

[Задание 3.19 8](#__RefHeading___Toc2853_3918689516)

# Постановка задачи

## 1. Списки (ввод с клавиатуры, использование рекурсии)

## Сформируйте новый список без повторений из тех элементов исходного списка, которые встречаются больше одного раза.

## Например, [5,1,2,1,3,5,5,7]-> [5,1].

## 2. Строки, файлы

## В текстовом файле, состоящем из нескольких строк, найдите все одинаковые слова и их количество и информацию об этом занесите в отдельную строку нового файла в виде: - повторов Строки в новом файле не должны повторяться.

**3. Динамические базы данных**

Создайте базу данных о сотрудниках: фамилия, имя, должность, оклад. Сформируйте список сотрудников с окладом ниже среднего по предприятию.

# 

# Ход работы

## Используемые функции

### Отдельные функции для 1.19

Watch – рекурсивно считывает каждый элемент и считает количество его повторений в списке.

finList – рекурсивно находит в списке значения которые повторяются больше одного раза.

### Отдельные функции для 2.19

check — рекурсивно получает все слова из файла в список.

Watch — рекурсивно получает максимальную длину строки в списке.

outList – выводит получившие значения в файл в порядке: элемент – кол-во повторений элемента.

### Отдельные функции для 3.19

### Mean – считает средний оклад. Wearble – суммирует оклад всех сотрудников. goal – выводит сотрудников, у которых оклад меньше среднего веса.

### Использованные встроенные функции

length(L,N) – подсчет количества элементов N в списке L.

delete - удаление всех вхождений заданного элемента.

read(A)- чтение значения с клавиатуры в переменную А.

write(A)- вывод значения А на экран без перевода строки.

read\_stream\_to\_codes(F,L) — Читает содержимое из входного потока F (до конца файла) и преобразует его в список кодов символов (включая коды перевода строки 10).

See, tell – открывает и перенаправляет ввод в файл.

told - закрытие файлов, открытых с помощью tell.

seeing – Связывает F с именем файла, являющегося текущим входным или выходным потоком.

open — открывает файл для работы с указанным доступом.

string\_to\_list - двунаправленное преобразование между строкой и списком кодов символов.

split\_string(S,R,D,L) - преобразует строку S в список подстрок L, используя R как разделитель, удаляя из начала и конца подстрок символы строки D.

*assert()* — Добавление в конец базы данных.

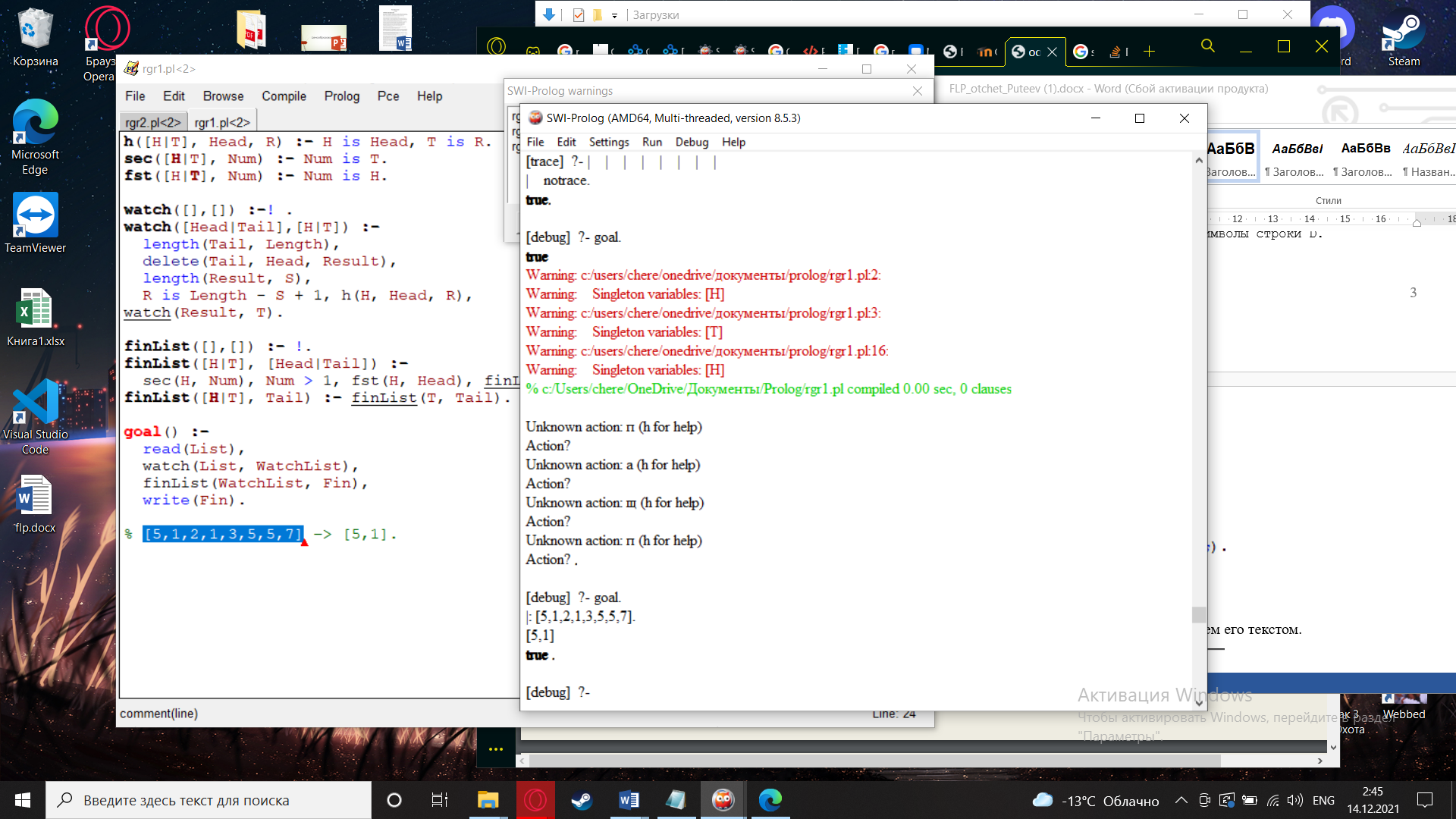
*retract(F)* — Удаление из динамической базы данных первого предложения, сопоставимого с *F*.

*retractall(F)* — Удаление из динамической базы данных всех предложений, сопоставимых с *F*. Предикат всегда успешен.

## Выполнение программы

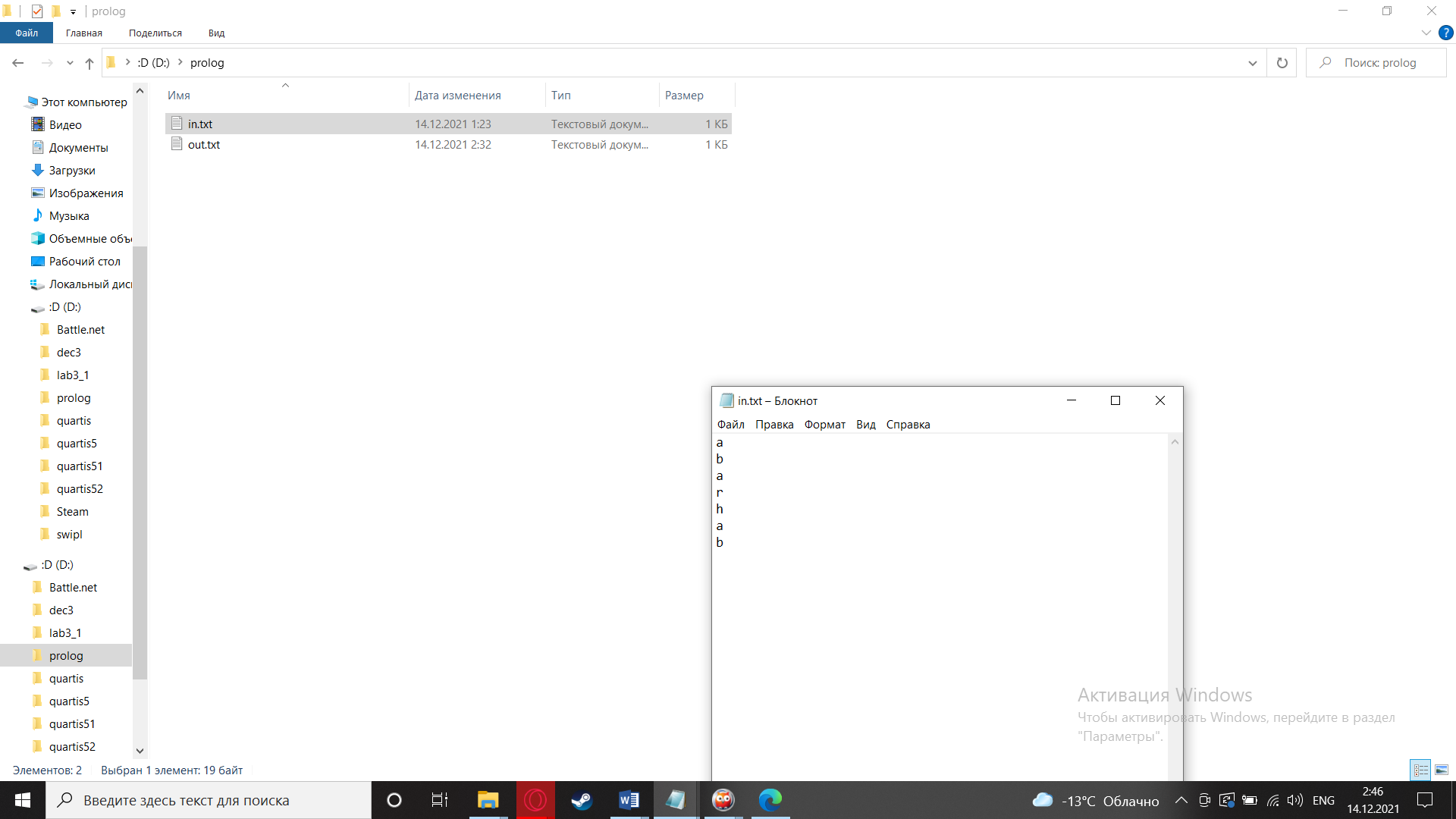
### Работа программы 1.19

### Результат работы программы.

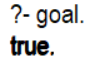


### Работа программы 2.19

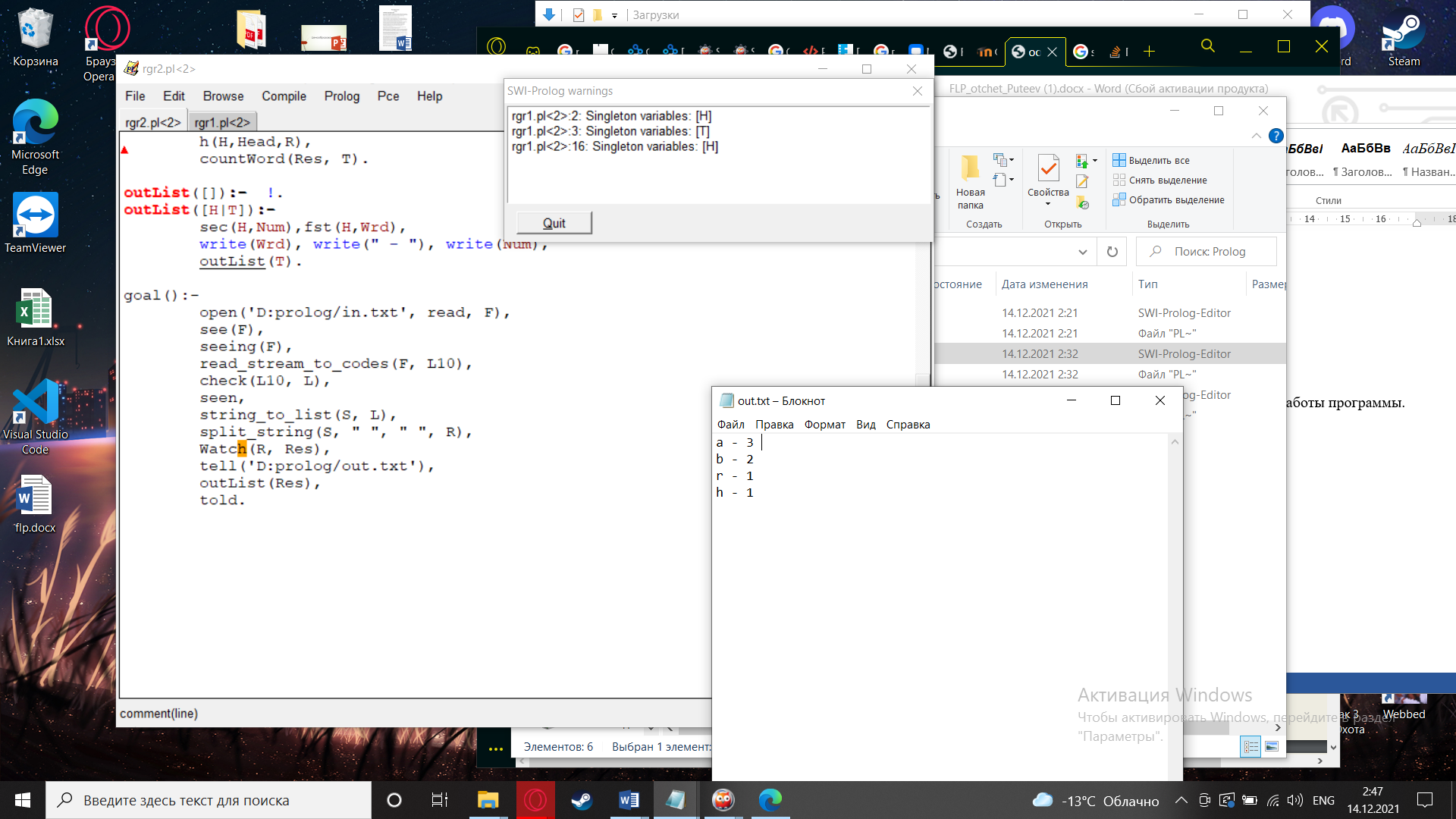
Создаём текстовый файл с названием «in.txt», заполняем его текстом.



Запускаем программу.

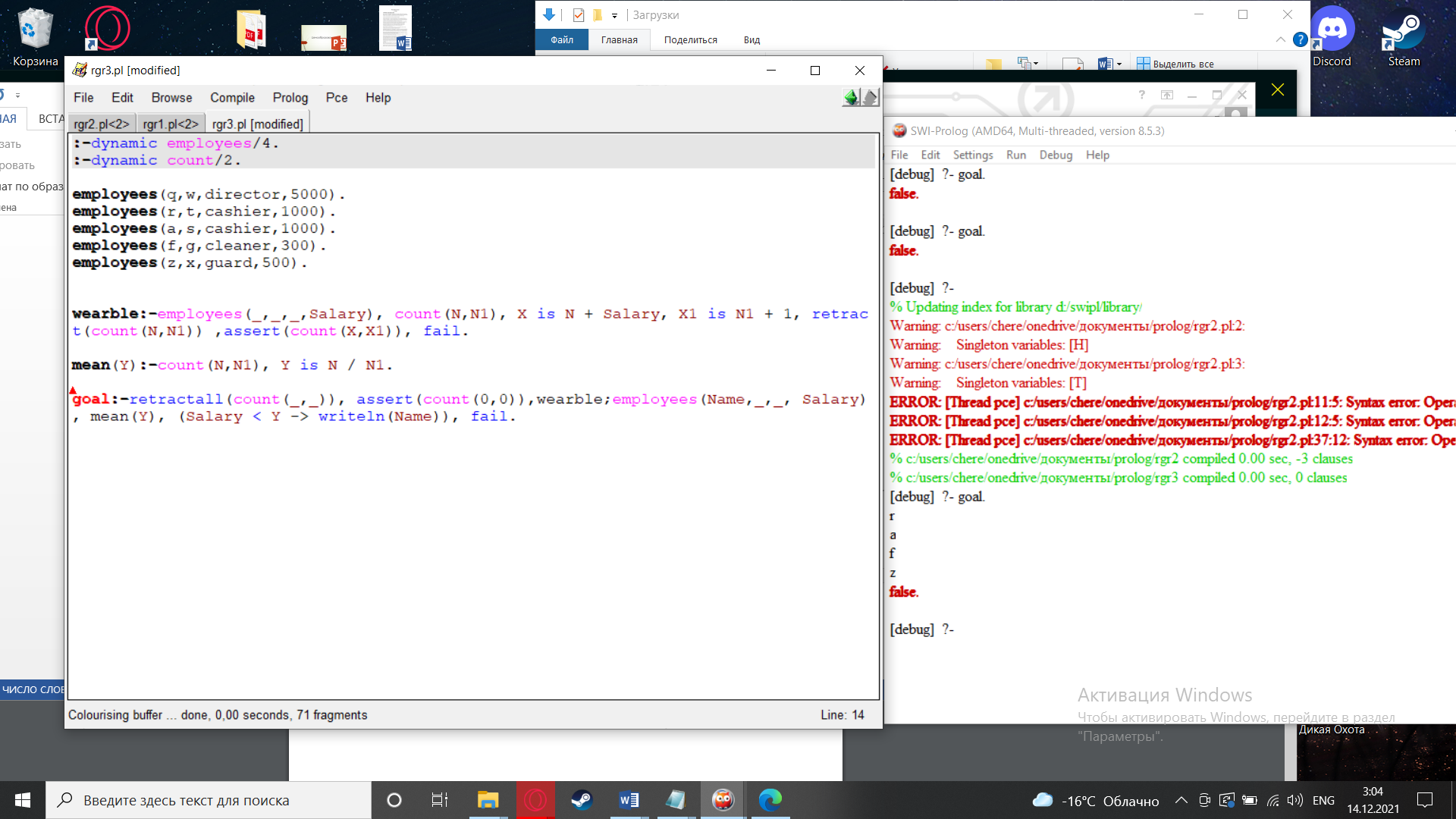


В результате появляется файл out.txt в котором записан результат работы программы.

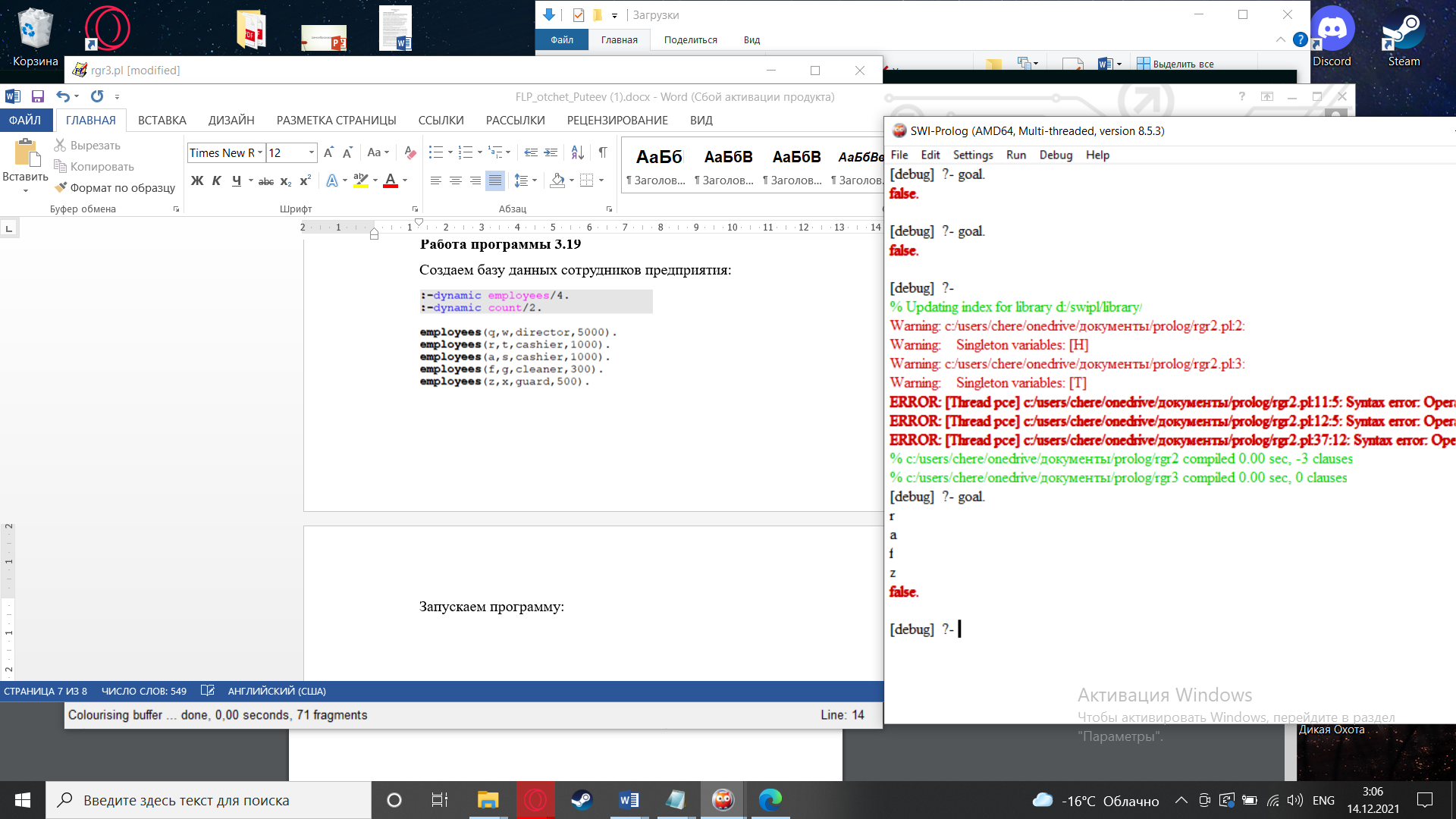


### Работа программы 3.19

Создаем базу данных сотрудников предприятия:



Запускаем программу:



# Список литературы

1. Википедия — свободная энциклопедия [Электронный ресурс]. URL: <https://ru.wikipedia.org/>. (Дата обращения: 29.11.2021).
2. SWI-Prolog [Электронный ресурс]. URL: <https://www.swi-prolog.org/>. (Дата обращения: 28.11.2021).
3. Biggest Online Tutorials Library [Электронный ресурс]. URL: <https://www.tutorialspoint.com/prolog/prolog_lists.htm/>. (Дата обращения: 29.11.2021).

# Листинг

## Задание 1.19

## h([H|T], Head, R) :- H is Head, T is R.

## sec([H|T], Num) :- Num is T.

## fst([H|T], Num) :- Num is H.

## watch([],[]) :-! .

## watch([Head|Tail],[H|T]) :-

## length(Tail, Length),

## delete(Tail, Head, Result),

## length(Result, S),

## R is Length - S + 1, h(H, Head, R),

## watch(Result, T).

## finList([],[]) :- !.

## finList([H|T], [Head|Tail]) :-

## sec(H, Num), Num > 1, fst(H, Head), finList(T, Tail).

## finList([H|T], Tail) :- finList(T, Tail).

## goal() :-

## read(List),

## watch(List, WatchList),

## finList(WatchList, Fin),

## write(Fin).

## Задание 2.19

h([H|T], Head, R) :- string\_chars(Head, H), T is R.

sec([H|T], Num) :- Num is T.

fst([H|T], Num) :- string\_chars(Str, H), Num = Str.

check([],[]):- !.

check([H|T],[Head|Tail]):-

H =:= 10, Head is 32, check(T,Tail).

check([H|T],[Head|Tail]):-

Head is H, check(T,Tail).

Watch([],[]):- !.

Watch([Head| Tail],[H|T]):-

length(Tail, Length),

%my\_delete(Tail, Head|Tail, Res),

delete(Tail, Head, Res),

%writeln(Res),

length(Res, Len),

R = Length - Len + 1,

h(H,Head,R),

countWord(Res, T).

outList([]):- !.

outList([H|T]):-

sec(H,Num),fst(H,Wrd),

write(Wrd), write(" - "), write(Num),

outList(T).

goal():-

open('D:prolog/in.txt', read, F),

see(F),

seeing(F),

read\_stream\_to\_codes(F, L10),

check(L10, L),

seen,

string\_to\_list(S, L),

split\_string(S, " ", " ", R),

Watch(R, Res),

tell('D:prolog/out.txt'),

outList(Res),

told.

## Задание 3.19

:-dynamic employees/4.

:-dynamic count/2.

employees(q,w,director,5000).

employees(r,t,cashier,1000).

employees(a,s,cashier,1000).

employees(f,g,cleaner,300).

employees(z,x,guard,500).

wearble:-employees(\_,\_,\_,Salary), count(N,N1), X is N + Salary, X1 is N1 + 1, retract(count(N,N1)) ,assert(count(X,X1)), fail.

mean(Y):-count(N,N1), Y is N / N1.

goal:-retractall(count(\_,\_)), assert(count(0,0)),wearble;employees(Name,\_,\_, Salary), mean(Y), (Salary < Y -> writeln(Name)), fail.