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 13

Выполнил:

студент группы ИП-715

Новосад Дарья Сергеевна

Работу проверила: ассистент кафедры ПмиК

Василина Валерьевна Долгополова

Новосибирск 2019 г.

**Содержание**

[1. Задание № 1. Списки. 3](#_Toc24299534)

[1.1. Условие задачи. 3](#_Toc24299535)

[1.2. Алгоритм решения поставленной задачи. 3](#_Toc24299536)

[1.3. Листинг программы. 3](#_Toc24299537)

[1.3 Скриншоты работы программы. 3](#_Toc24299538)

[2. Задание № 2. Строки, файлы. 3](#_Toc24299539)

[2.1 Условие задачи. 3](#_Toc24299540)

[2.2. Алгоритм решения поставленной задачи. 4](#_Toc24299541)

[2.3. Листинг программы. 4](#_Toc24299542)

[2.4. Скриншоты программы. 4](#_Toc24299543)

# **1. Задание № 1. Списки.**

## **1.1. Условие задачи.**

Удалите из списка все вхождения элемента X. Например, [1,2,1,3,1,1,2,5,6], X=1-> [2,3,2,5,6].

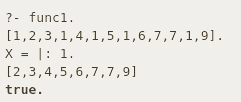
## **1.2. Алгоритм решения поставленной задачи.**

Вводится список, из которого необходимо удалить все вхождения элемента Х, вводится сам элемент Х. С помощью встроенного предиката *delete*(<список, из которого необходимо удалить все вхождения элемента>, <значение элемента, которое нужно>, <список без элемента и его повторений>). Выводится результирующий список, и программа завершается.

## **1.3. Листинг программы.**

|  |
| --- |
| func1:- |
|  | write('Список: '), |
|  | read(List), |
|  | write('X = '), |
|  | read(X), |
|  | delete(List,X,L1), |
|  | write(L1). |

## **1.3 Скриншоты работы программы.**

****

# **2. Задание № 2. Строки, файлы.**

## **2.1 Условие задачи.**

Переставьте строки текстового файла в обратном порядке. Сформируйте новый файл.

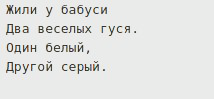
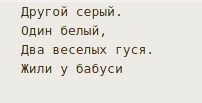
## **2.2. Алгоритм решения поставленной задачи.**

Через предикат *open* открываются два файла: первый на чтение, второй на запись. Потоки вывода и ввода привязываются к соответствующим файлам (для чтения и записи). Файловая переменная передается в функцию, в которой происходит построчное чтение из файла и данные в виде кодов заносятся в список (встроенный предикат *read\_line\_to\_codes*). Затем, с помощью предиката *name* список из кодов символов преобразуется в строку. Пока не встречен конец файла (*at\_end\_of\_stream*) вновь возвращаемся в нашу функцию. На обратном ходу рекурсии полученные строки записываются в файл, в результате чего, получается обратный порядок строк.

## **2.3. Листинг программы.**

|  |
| --- |
| r(F1):- |
|  | read\_line\_to\_codes(F1,L), |
|  | name(S,L), |
|  | (at\_end\_of\_stream, write('');r(F1),!), |
|  | writeln(S). |
|  |  |
|  | func2:- |
|  | open('text.txt',read,F1), |
|  | open('result.txt',write,F2), |
|  | set\_input(F1), |
|  | set\_output(F2), |
|  | r(F1), |
|  | close(F1), |
|  | close(F2). |

## **2.4. Скриншоты программы.**

** **