

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 ИВ222

Рудцких В.Е.
ФИО студента

Работу проверил: Сороковых Д.А.
ФИО преподавателя

Задание

Новосибирск 2024 г.

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

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

3.13. Создайте базу данных об игрушках: название, стоимость. Получите названия наиболее дешевых игрушек (цены которых отличаются от самой дешевой не более чем на 100 рублей).

Листинг

1.13:

```
del([], _, []).
```

```
del([H|T], H, Res) :- !, del(T, H, Res).
```

```
del([H|T], X, [H|Res]) :- del(T, X, Res).
```

```
rgr1:- writeln("Введите список "), read(List), writeln("Введите удаляемый элемент "), read(D), del(List,  
D, List_new), writeln("Результат "),writeln(List_new).
```

2.13:

```
reverseLinesOfFile(In, Out) :- repeat, read_line_to_string(In, S), (S == end_of_file, !;  
reverseLinesOfFile(In, Out), writeln(Out, S), fail).
```

```
print_reverseLinesOfFile :- format('nВведите название входного файла (Файл должен находиться в  
директории программы и иметь расширение .txt) ~n'), read(I), format('nВведите название  
выходного файла (Файл будет находиться в директории программы и иметь расширение .txt) ~n'),  
read(O), concat(I, '.txt', Input), concat(O, '.txt', Output), open(Input, read, In), open(Output, write, Out),  
reverseLinesOfFile(In, Out), close(In), close(Out).
```

```
rgr2 :- print_reverseLinesOfFile.
```

3.13:

`:- dynamic toy/2.`

`% Инициализация базы данных из файла load_database :- exists_file('toys.db'), retractall(toy(_, _)),
see('toys.db'), read_toys, seen.`

`read_toys :- read(Toy), (Toy == end_of_file -> true ; assertz(Toy), read_toys).`

`% Сохранение базы данных в файл save_database :- tell('toys.db'), listing(toy/2), told.`

`% Просмотр содержимого базы данных view_db :- writeln('Содержимое базы данных:'),
forall(toy(Name, Price), format('w~w рублей\n', [Name, Price])).`

`add_toys :- writeln('Введите название и стоимость игрушек (в формате "Название Стоимость"). Для
завершения введите "end".'), repeat, read_line_to_string(user_input, Input), (Input == "end" -> ! %
завершает repeat, возвращает в меню ; split_string(Input, " ", "", [NameStr, PriceStr]),
atom_string(Name, NameStr), atom_number(PriceStr, Price), assertz(toy(Name, Price)), % добавление
игрушки fail % продолжает цикл).`

`% Удаление записей remove_toys :- writeln('Введите название игрушек для удаления (в формате
"Название1 Название2 ..."). Для завершения введите "end".'), repeat,
read_line_to_string(user_input, Input), (Input == "end" -> ! ; split_string(Input, " ", "", Names),
forall(member(NameStr, Names), (atom_string(Name, NameStr), retractall(toy(Name, _)))), fail).`

`% Выполнение запроса query_cheapest :- findall(Price, toy(_, Price), Prices), min_list(Prices, MinPrice),
MaxPrice is MinPrice + 100, writeln('Самые дешевые игрушки:'), forall((toy(Name, Price), Price =<
MaxPrice, Price >= MinPrice), format('w~w рублей\n', [Name, Price])).`

`menu :- writeln('Главное меню:'), writeln('1. Просмотр содержимого базы данных'), writeln('2.
Добавление новых игрушек'), writeln('3. Удаление записей'), writeln('4. Поиск самых дешевых
игрушек'), writeln('5. Выход'), read(Choice), handle_choice(Choice), menu. % возвращается обратно в
меню`

`handle_choice(1) :- view_db. handle_choice(2) :- add_toys. handle_choice(3) :- remove_toys.
handle_choice(4) :- query_cheapest. handle_choice(5) :- save_database, writeln('Выход из
программы.'), halt. handle_choice(_) :- writeln('Неверный выбор, попробуйте еще раз.'), fail.`

`% Запуск программы start :- load_database, menu.`

Пример toys.db

`:- dynamic toy/2.`

`toy('Кукла', 450).`

`toy('Паровозик', 700).`

toy('Скакалка', 400).

toy('Кубики', 510).

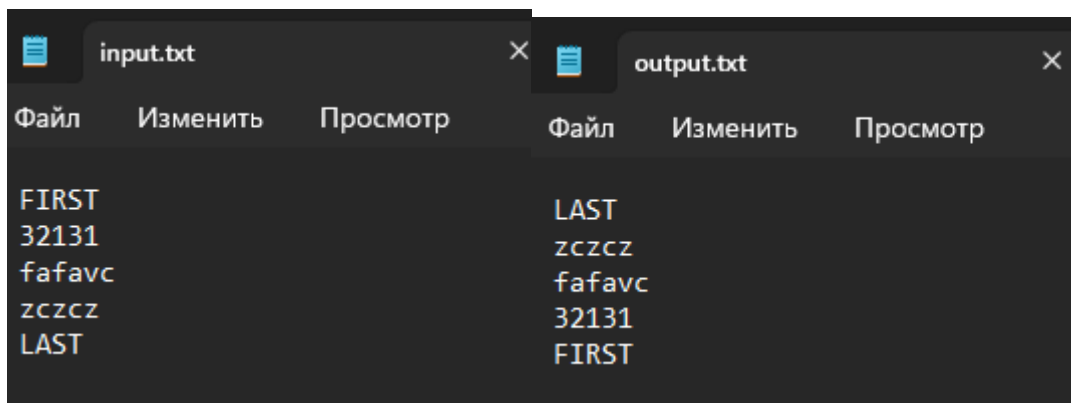
Тест

1.13:

```
?- rgr1.  
Введите список  
|: [1,2,3,4,4,1,1,5,9,8,11].  
Введите удаляемый элемент  
|: 1.  
Результат  
[2,3,4,4,5,9,8,11]  
true.  
  
?-
```

2.13:

```
?- rgr2.  
Введите название входного файла (файл должен находиться в директории программы и иметь расширение .txt)  
|: input.  
  
Введите название выходного файла (файл будет находиться в директории программы и иметь расширение .txt)  
|: output.  
  
true.  
  
?- █
```



3.13:

SWI-Prolog (AMD64, Multi-threaded, version 9.2.8)

File Edit Settings Run Debug Help

```
?- start.
Главное меню:
1. Просмотр содержимого базы данных
2. Добавление новых игрушек
3. Удаление записей
4. Поиск самых дешевых игрушек
5. Выход
|: 1.
Содержимое базы данных:
Кукла - 450 рублей
Паровозик - 700 рублей
Скакалка - 400 рублей
Кубики - 510 рублей
Главное меню:
1. Просмотр содержимого базы данных
2. Добавление новых игрушек
3. Удаление записей
4. Поиск самых дешевых игрушек
5. Выход
|: 2.
Введите название и стоимость игрушек (в формате "Название Стоимость"). Для завершения введите "end".
|: Солдатик 420
|: Машина 430
|: end
Главное меню:
1. Просмотр содержимого базы данных
2. Добавление новых игрушек
3. Удаление записей
4. Поиск самых дешевых игрушек
5. Выход
|: 4.
Самые дешевые игрушки:
Кукла - 450 рублей
Скакалка - 400 рублей
Солдатик - 420 рублей
Машина - 430 рублей
Главное меню:
1. Просмотр содержимого базы данных
2. Добавление новых игрушек
3. Удаление записей
4. Поиск самых дешевых игрушек
5. Выход
|: 3.
Введите название игрушек для удаления (в формате "Название1 Название2 ..."). Для завершения введите "end".
|: Кукла
|: end
Главное меню:
1. Просмотр содержимого базы данных
2. Добавление новых игрушек
3. Удаление записей
4. Поиск самых дешевых игрушек
5. Выход
|: 1.
Содержимое базы данных:
Паровозик - 700 рублей
Скакалка - 400 рублей
```