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 9

Выполнил:

студент группы ИС-142 Наумов А.А.
ФИО студента

Работу проверил: Сороковых Д.А.
ФИО преподавателя

Новосибирск 2023 г.

ЗАДАНИЕ

1. По кругу расположены n человек. Начиная с некоторой позиции, считаем от 1 до m по кругу. Каждый m-ый человек выбывает из круга, при этом круг смыкается. Определите порядок выбывания из круга. Например, если n = 7, m = 3, то порядок выбывания будет таким: 3, 6, 2, 7, 5, 1, 4.
2. В текстовом файле, состоящем из нескольких строк, найдите слова, содержащие наибольшее число гласных букв (a,e,i,o,u). Сформируйте новый файл из найденных слов.

РЕШЕНИЕ

Задание 1.9

Общие требования:
1. Назначение программы: Программа предназначена для решения задачи Иосифа Флавия, алгоритма, который определяет порядок выбывания людей из круга при заданных начальных условиях.
2. Платформа выполнения: Prolog (SWI-Prolog или совместимый).

Функциональные требования:
1. Ввод данных: Программа должна принимать два числа: количество людей в кругу (N) и шаг счёта (M).
2. Обработка данных: Программа должна создать начальный список людей, пронумерованных от 1 до N, и затем, начиная с первого человека в списке, удалять каждого M-го человека из списка до тех пор, пока в списке не останется ни одного человека.
3. Вывод результатов: Программа должна выводить порядок, в котором люди были удалены из списка.

Нетехнические требования:
1. Документация и комментарии: Код должен быть хорошо документирован с подробными комментариями, объясняющими ключевые элементы и логику работы программы.
2. Простота и читаемость кода: Код должен быть написан с учетом принципов чистого кода для удобства чтения и дальнейшего расширения.

Тестирование:
1. Сценарии тестирования: Программа должна быть протестирована на различных входных данных, чтобы убедиться в её корректной работе. Например, тесты для наборов входных данных (N=7, M=3), (N=10, M=2) и т.д.

Доставка:
1. Формат поставки: Исходный код программы должен быть предоставлен в формате, совместимом с SWI-Prolog.
2. Инструкции по запуску: Загрузить программу в prolog и запустить команду.

Код для решения задачи Иосифа Флавия в Prolog работает следующим образом:

1. Создание начального списка:
   - Предикат `create_list(N, List)` используется для создания списка, содержащего числа от 1 до `N`. Эти числа представляют людей, стоящих в кругу.

2. Основной алгоритм:
   - Предикат `josephus(N, M, Order)` является точкой входа. Он принимает `N` (общее количество людей), `M` (шаг счёта), и переменную `Order`, в которую будет записан итоговый порядок выбывания людей.
   - Внутри `josephus`, сначала вызывается `create_list` для создания начального списка людей, а затем вызывается вспомогательный предикат `josephus_step` для выполнения основной логики алгоритма.

3. Выполнение шагов алгоритма:
   - `josephus_step(List, M, Order)` запускает процесс с первого элемента в списке и итеративно удаляет каждого `M`-го человека.
   - Внутри `josephus_step`, предикат `josephus_cycle` вызывается рекурсивно для обработки оставшихся элементов списка после каждого удаления.
   - `josephus_cycle(List, M, Start, Acc, Order)` принимает текущий список (`List`), шаг счёта (`M`), начальную позицию (`Start`), аккумулятор для сохранения порядка удаления (`Acc`), и итоговый порядок (`Order`).

4. Логика удаления элементов:
   - В `josephus_cycle`, вычисляется позиция удаляемого элемента с учётом текущей длины списка и начальной позиции.
   - Используется предикат `nth0(Pos, List, Removed, Rest)`, который удаляет элемент из списка. `Pos` — позиция удаляемого элемента, `Removed` — удаляемый элемент, `Rest` — оставшийся список после удаления.
   - После каждого удаления, удалённый элемент добавляется в аккумулятор `Acc`, и процесс продолжается с оставшегося списка.

5. Завершение и вывод результатов:
   - Как только весь список обработан (все элементы удалены), порядок выбывания, сохранённый в аккумуляторе, переворачивается (так как удаление шло с конца списка к началу), и результат возвращается в переменную `Order`.

Этот алгоритм эффективно моделирует процесс задачи Иосифа Флавия, последовательно удаляя каждого `M`-го человека из круга до тех пор, пока не останется никого.
Листинг программы:

```
% Предикат для создания начального списка
create_list(N, List) :- findall(Num, between(1, N, Num),
List).

% Предикат для выполнения алгоритма Иосифа Флавия
josephus(N, M, Order) :-
    create_list(N, List),
    josephus_step(List, M, Order).

% Шаг алгоритма Иосифа Флавия
josephus_step(List, M, Order) :-
    josephus_cycle(List, M, 0, [], RevOrder),
    reverse(RevOrder, Order).

josephus_cycle([], _, _, Acc, Acc).
josephus_cycle(List, M, Start, Acc, Order) :-
    length(List, Len),
    Pos is (Start + M - 1) mod Len,
    nth0(Pos, List, Removed, Rest),
    josephus_cycle(Rest, M, Pos, [Removed|Acc], Order).

% Пример использования: josephus(7, 3, Order).
```

Скриншот работы:

```
% /home/alexeynaumov/prolog/lab.pl compiled 0.00 sec, 10 clauses
?- josephus(7, 3, Order).
Order = [3, 6, 2, 7, 5, 1, 4]
```

Задание 2.9

Общее описание

Разработать программу на языке Prolog, которая читает текстовый файл, анализирует каждую строку для нахождения слов с максимальным количеством гласных букв (a, e, i, o, u) и записывает эти слова в новый текстовый файл.

Функциональные требования

1. Чтение файла: Программа должна уметь читать текстовые файлы. Каждая строка файла должна обрабатываться отдельно.
2. Обработка текста: Программа должна разбивать каждую строку на слова и подсчитывать количество гласных букв в каждом слове. Гласные буквы включают 'a', 'e', 'i', 'o', 'u' в любом регистре.

3. Выбор слов: В каждой строке должны быть выбраны слова с наибольшим количеством гласных букв. Если в строке несколько слов с одинаковым максимальным количеством гласных, выбираются все эти слова.
4. Запись в файл: Выбранные слова должны быть записаны в новый текстовый файл. Каждая строка выходного файла должна соответствовать строке входного файла и содержать выбранные слова, разделённые пробелами.

Нетехнические требования
1. Документация: Код должен быть хорошо документирован, с комментариями, объясняющими основные части программы.
2. Удобство использования: Программа должна быть простой в использовании, с чёткими инструкциями по запуску и вводу данных.

Тестирование
Программа должна быть протестирована на различных текстовых файлах для убеждения в корректности работы всех функций.

Доставка
Программа должна быть предоставлена в виде исходного кода на Prolog, сопровождаемого инструкциями по запуску и примерами использования.

Этот код предназначен для обработки текстового файла, чтобы найти и сохранить слова с наибольшим количеством гласных букв на строку. Давайте рассмотрим, как он работает на примере файла `input.txt`.

1. Чтение файла (`read_file_to_lines`):
   - Процесс начинается с чтения файла `input.txt`. Каждая строка файла считывается и сохраняется в список `Lines`.
2. Обработка каждой строки (`words_with_most_vowels` и `lines_with_most_vowels`):
   - Для каждой строки из списка `Lines` выполняется функция `lines_with_most_vowels`. Эта функция разделяет строку на слова (используя `split_string`) и затем для каждого слова подсчитывает количество гласных (`vowel_count`).
   - После подсчета гласных находится максимальное количество гласных в строке (`max_list`). Затем выбираются все слова, имеющие это максимальное количество гласных.

3. Запись результатов в файл (`write_lines_to_file` и `write_line`):
   - Найденные слова для каждой строки объединяются в одну строку (с использованием `atomic_list_concat`) и записываются в новый файл, `output.txt`.

Рассмотрим пример работы этого кода на тексте:

- **Первая строка**: "This is a sample text file."

- Слова "sample" и "file" имеют по 2 гласные, что является максимумом для этой строки.
- **Вторая строка**:"It contains several lines of text."
- Слова "contains" и "several" имеют по 3 гласные, что является максимумом для этой строки.
- **Третья строка**:"The program should find words with the maximum number of vowels."
- Слово " maximum " имеет 3 гласные, что является максимумом для этой строки.

В результате, `output.txt` будет содержать следующие строки:
- "sample file"
- "contains several"
- "maximum"

Листинг программы

```
% Предикат для чтения файла в список строк
read_file_to_lines(FileName, Lines) :-
    open(FileName, read, Stream),
    read_lines(Stream, Lines),
    close(Stream).

read_lines(Stream, Lines) :-
    read_line_to_string(Stream, Line),
    Line \= end_of_file,
    !,
    read_lines(Stream, RestLines),
    Lines = [Line | RestLines].
read_lines(_, []).

% Предикат для подсчета гласных в слове
vowel_count(Word, Count) :-
    string_lower(Word, LowerWord),
    string_chars(LowerWord, Chars),
    include(is_vowel, Chars, Vowels),
    length(Vowels, Count).

is_vowel(Char) :-
    member(Char, ['a', 'e', 'i', 'o', 'u']).

% Измененный предикат для выбора всех слов с наибольшим
количеством гласных
words_with_most_vowels(Lines, Words) :-
    maplist(lines_with_most_vowels, Lines, Words).
```

```
lines_with_most_vowels(Line, Words) :-
    split_string(Line, " ", "", SplitLine),
    maplist(vowel_count, SplitLine, Counts),
    max_list(Counts, Max),
    findall(Word, (member(Word, SplitLine),
vowel_count(Word, Max)), Words).

% Измененный предикат для записи списков слов в файл
write_lines_to_file(FileName, Lines) :-
    open(FileName, write, Stream),
    maplist(write_line(Stream), Lines),
    close(Stream).

write_line(Stream, Words) :-
    atomic_list_concat(Words, ' ', Line),
    writeln(Stream, Line).

% Основной предикат для выполнения задачи
process_files(InputFile, OutputFile) :-
    read_file_to_lines(InputFile, Lines),
    words_with_most_vowels(Lines, WordsList),
    write_lines_to_file(OutputFile, WordsList).



%
process_files('/home/alexeynaumov/prolog/input.txt','/home
/alexeynaumov/prolog/output.txt').
```

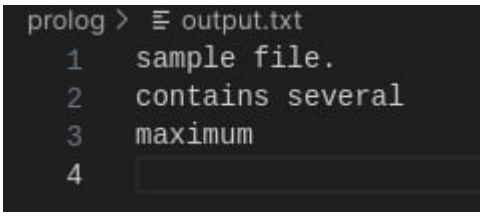
Скриншот работы и подаваемых файлов:

```
% /home/alexeynaumov/prolog/lab.pl compiled 0.00 sec, 10 clauses
?- process_files('/home/alexeynaumov/prolog/input.txt','/home/alexeynaumov/prolog/
output.txt').
true.

?-
```

prolog > input.txt

```
1  This is a sample text file.
2  It contains several lines of text.
3  The program should find words with the maximum number of vowels.
4
```


prolog > output.txt
1 sample file.
2 contains several
3 maximum
4