

## **Вариант 1**

Написать программу, которая в качестве аргумента командной строки принимает имя текстового файла, содержащего слова. Необходимо вывести на стандартный выходной поток список всех различных слов в файле в порядке возрастания количества вхождений слов (с указанием количества вхождений). При реализации алгоритма необходимо использовать односвязный список, каждый узел которого должен содержать указатель на слово и количество вхождений этого слова. Слова во входном файле разделяются символами, для которых библиотечные функции isspace() или ispunct() возвращают ненулевое значение.

## **Вариант 2**

Написать программу, которая в качестве аргумента командной строки принимает имя текстового файла, содержащего слова. Необходимо вывести на стандартный выходной поток список всех различных слов в файле в порядке убывания количества вхождений слов (с указанием количества вхождений). При реализации алгоритма необходимо использовать односвязный список, каждый узел которого должен содержать указатель на слово и количество вхождений этого слова. Слова во входном файле разделяются символами, для которых библиотечные функции isspace() или ispunct() возвращают ненулевое значение.

### **Вариант 3**

Написать программу, которая в качестве аргумента командной строки принимает имя текстового файла, содержащего слова. Необходимо вывести на стандартный выходной поток список всех различных слов в файле в порядке возрастания количества вхождений слов (с указанием количества вхождений). При реализации алгоритма необходимо использовать двусвязный список, каждый узел которого должен содержать указатель на слово и количество вхождений этого слова. Слова во входном файле разделяются символами, для которых библиотечные функции isspace() или ispunct() возвращают ненулевое значение.

## **Вариант 4**

Написать программу, которая в качестве аргумента командной строки принимает имя текстового файла, содержащего слова. Необходимо вывести на стандартный выходной поток список всех различных слов в файле в порядке убывания количества вхождений слов (с указанием количества вхождений). При реализации алгоритма необходимо использовать двусвязный список, каждый узел которого должен содержать указатель на слово и количество вхождений этого слова. Слова во входном файле разделяются символами, для которых библиотечные функции isspace() или ispunct() возвращают ненулевое значение.

## **Вариант 5**

Написать программу, которая в качестве аргумента командной строки принимает имя текстового файла, содержащего строку символов, в которой записано выражение со скобками трёх типов: [ ], { }, ( ). Необходимо определить, правильно ли расставлены скобки (не учитывая остальные символы) и напечатать на стандартный выходной поток результат. При реализации алгоритма необходимо использовать стек. Например, в выражении [()]{ } скобки расставлены правильно, а в выражениях ][ и [{()}]} неправильно.

## **Вариант 6**

Написать программу, которая в качестве аргумента командной строки принимает имя текстового файла, содержащего строку символов, представляющую собой арифметическое выражение. Выражение записано в постфиксной форме (знаки операций идут после операндов с учётом приоритетов). Выражение правильное, допускаются только однозначные числа и знаки операций +, -, \*, /. Необходимо вычислить значение выражения и напечатать на стандартный выходной поток результат. При реализации алгоритма необходимо использовать стек.

## **Вариант 7**

Написать программу, которая в качестве аргументов командной строки принимает имена двух текстовых файлов. В первом файле содержится последовательность целых чисел. Необходимо записать во второй файл числа, содержащиеся в первом файле, изменив порядок их следования на обратный. При реализации алгоритма необходимо использовать динамический список. Числа в первом файле разделяются символами, для которых библиотечная функция `isspace()` возвращает ненулевое значение.

## **Вариант 8**

Написать программу, которая в качестве аргументов командной строки принимает имена двух текстовых файлов. В первом файле содержится последовательность слов. Необходимо записать во второй файл слова, содержащиеся в первом файле, изменив порядок их следования на обратный. При реализации алгоритма необходимо использовать динамический список. Слова в первом файле разделяются символами, для которых библиотечная функция `isspace()` возвращает ненулевое значение.

## **Вариант 9**

Написать программу, которая в качестве аргументов командной строки принимает имена двух текстовых файлов. В первом файле содержится последовательность целых чисел. Необходимо записать во второй файл числа, содержащиеся в первом файле, упорядоченные в порядке возрастания. При реализации алгоритма необходимо использовать динамический список. Числа в первом файле разделяются символами, для которых библиотечная функция `isspace()` возвращает ненулевое значение.

## **Вариант 10**

Написать программу, которая в качестве аргументов командной строки принимает имена двух текстовых файлов. В первом файле содержится последовательность слов. Необходимо записать во второй файл слова, содержащиеся в первом файле, упорядоченные в порядке убывания (в качестве функции сравнения слов использовать лексикографическое сравнение). При реализации алгоритма необходимо использовать динамический список. Слова в первом файле разделяются символами, для которых библиотечная функция `isspace()` возвращает ненулевое значение.

## **Вариант 11**

Написать программу, которая в качестве аргументов командной строки принимает имена двух текстовых файлов. В первом файле содержится последовательность целых чисел. Необходимо записать во второй файл числа, содержащиеся в первом файле, упорядоченные в порядке возрастания. При реализации алгоритма необходимо использовать двоичное дерево поиска (ключами при построении дерева должны являться сами числа). Числа в первом файле разделяются символами, для которых библиотечная функция `isspace()` возвращает ненулевое значение.

## **Вариант 12**

Написать программу, которая в качестве аргументов командной строки принимает имена двух текстовых файлов. В первом файле содержится последовательность слов. Необходимо записать во второй файл слова, содержащиеся в первом файле, упорядоченные в порядке убывания. При реализации алгоритма необходимо использовать двоичное дерево поиска (ключами при построении дерева должны являться сами слова, в качестве функции сравнения слов использовать лексикографическое сравнение). Слова в первом файле разделяются символами, для которых библиотечная функция `isspace()` возвращает ненулевое значение.

## **Вариант 13**

Написать программу, которая в качестве аргумента командной строки принимает имя текстового файла, содержащего элементы трёх видов:

- + <неотрицательное целое число>
- <неотрицательное целое число>
- ? <неотрицательное целое число>

Элементы отделяются друг от друга одним или несколькими разделителями – пробелами, табуляциями, символами новой строки. Число с предшествующим плюсом добавляется в двоичное дерево поиска, если его там ещё нет (ключом при построении дерева должно являться само число). Если числу предшествует минус, то это число удаляется из дерева (если оно было в нём). Если перед числом стоит вопрос, то оно печатается в выходной поток в отдельной строке вместе со словом Yes или No в зависимости от того, присутствует ли это число в построенном на тот момент дереве поиска.

## **Вариант 14**

Написать программу, которая в качестве аргумента командной строки принимает имя текстового файла, содержащего элементы трёх видов:

+ <слово>  
- <слово>  
? <слово>

Элементы отделяются друг от друга одним или несколькими разделителями – пробелами, табуляциями, символами новой строки. Слово с предшествующим плюсом добавляется в двоичное дерево поиска, если его там ещё нет (ключом при построении дерева должно являться само слово, в качестве функции сравнения слов использовать лексикографическое сравнение). Если числу предшествует минус, то это слово удаляется из дерева (если оно было в нём). Если перед словом стоит вопрос, то оно печатается в выходной поток в отдельной строке вместе со словом Yes или No в зависимости от того, присутствует ли это слово в построенном на тот момент дереве поиска.

## **Вариант 15**

Написать программу, которая в качестве аргумента командной строки принимает имя текстового файла, содержащего элементы трёх видов:

+ <слово>  
- <слово>  
? <слово>

Элементы отделяются друг от друга одним или несколькими разделителями – пробелами, табуляциями, символами новой строки. Слово с предшествующим плюсом добавляется в упорядоченный динамический список, если его там ещё нет (в качестве функции сравнения слов использовать лексикографическое сравнение). Если числу предшествует минус, то это слово удаляется из списка (если оно было в нём). Если перед словом стоит вопрос, то оно печатается в выходной поток в отдельной строке вместе со словом Yes или No в зависимости от того, присутствует ли это слово в построенном на тот момент списке.

## **Вариант 16**

Написать программу, которая в качестве аргумента командной строки принимает имя текстового файла, содержащего элементы трёх видов:

- + <неотрицательное целое число>
- <неотрицательное целое число>
- ? <неотрицательное целое число>

Элементы отделяются друг от друга одним или несколькими разделителями – пробелами, табуляциями, символами новой строки. Число с предшествующим плюсом добавляется в упорядоченный динамический список, если его там ещё нет. Если числу предшествует минус, то это число удаляется из списка (если оно было в нём). Если перед числом стоит вопрос, то оно печатается в выходной поток в отдельной строке вместе со словом Yes или No в зависимости от того, присутствует ли это число в построенном на тот момент списке.

## **Вариант 17**

Написать программу, которая в качестве аргумента командной строки принимает имя текстового файла, содержащего элементы трёх видов:

+ <слово>

- <слово>

? <слово>

Элементы отделяются друг от друга одним или несколькими разделителями – пробелами, табуляциями, символами новой строки. Слово с предшествующим плюсом добавляется в упорядоченный двусвязный список, если его там ещё нет (в качестве функции сравнения слов использовать лексикографическое сравнение). Если числу предшествует минус, то это слово удаляется из двусвязного списка (если оно было в нём). Если перед словом стоит вопрос, то оно печатается в выходной поток в отдельной строке вместе со словом Yes или No в зависимости от того, присутствует ли это слово в построенном на тот момент списке.