

## Требования к программам

1. Задачи оцениваются независимо в двух группах: задачи 1–5 и задачи 6–9.
2. Программа должна получать все параметры в качестве аргументов командной строки.
3. Аргументы командной строки для задачи 1:

- 1)  $f_{in}$  – имя входного файла,
- 2)  $f_{out}$  – имя выходного файла,
- 3)  $s$  – строка  $s$ ,
- 4)  $t$  – строка  $t$ ,
- 5)  $m$  – параметр  $m$  алгоритма поиска.

Например, запуск

```
./a01.out a.txt b.txt "abcd" "(){}[] ;+-*/*=" 2
```

означает, что требуется читать строки из файла `a.txt`, выводить результат в файл `b.txt`, параметр-строка  $s$  равен `"abcd"`, параметр-строка  $t$  равен `"(){}[] ;+-*/*="`, а параметр  $m = 2$ .

4. Аргументы командной строки для задач 2, 6–9:

- 1)  $f_{in}$  – имя входного файла,
- 2)  $f_{out}$  – имя выходного файла,
- 3)  $s$  – строка  $s$ ,
- 4)  $t$  – строка  $t$ .

Например, запуск

```
./a01.out a.txt b.txt "abcd" "(){}[] ;+-*/*="
```

означает, что требуется читать строки из файла `a.txt`, выводить результат в файл `b.txt`, параметр-строка  $s$  равен `"abcd"`, параметр-строка  $t$  равен `"(){}[] ;+-*/*="`.

5. Аргументы командной строки для задач 3–5:

- 1)  $f_{in}$  – имя входного файла,
- 2)  $f_{out}$  – имя выходного файла,
- 3)  $s$  – строка  $s$ ,
- 4)  $t$  – строка  $t$ ,
- 5)  $x$  – строка  $x$ .

Например, запуск

```
./a03.out a.txt b.txt "abcd" "(){}[] ;+-*/*=" "ABCD"
```

означает, что требуется читать строки из файла `a.txt`, выводить результат в файл `b.txt`, параметр-строка `s` равен "abcd", параметр-строка `t` равен "(){}[] ;+-\*/=", параметр-строка `x` равен "ABCD".

6. Результатом работы каждой функции является измененный файл (не выводится в `main`) и возвращаемое значение (выводится в `main`).
7. Вывод результата работы функции в функции `main` должен производиться по формату:

```
printf ("%s : Task = %d Result = %d Elapsed = %.2f\n",
        argv[0], task, res, t);
```

где

- `argv[0]` – первый аргумент командной строки (имя образа программы),
- `task` – номер задачи (1–9),
- `res` – результат работы функции, реализующей решение этой задачи,
- `t` – время работы функции, реализующей решение этой задачи.

**Вывод должен производиться в точности в таком формате**, чтобы можно было автоматизировать обработку запуска многих тестов.

### Задачи

1. Написать функцию, получающую в качестве аргументов имя `a` текстового файла неизвестной длины, имя файла `b` для вывода информации, символьные строки `s`, `t`, целое число `m`, указатель на целое число `r`, и выводящую в файл `b` те строки файла `a`, которые имеют общее слово со строкой `s`. Словом называется последовательность символов, не содержащая пробельных символов, пробельным называется символ, содержащийся в строке `t`. Для быстрого поиска все слова в строке `s` размещаются в В-дереве строк по основанию `m` до начала работы с текстовыми файлами. Сами строки и массивы в узлах В-дерева размещаются как объекты `std::unique_ptr<тип>`. Каждая прочитанная из файла строка разбирается на слова, которые затем ищутся в В-дереве. Функция возвращает в случае успеха количество выведенных в файл `b` строк в переменной `r`. Возвращаемое значение функции имеет тип `io_status` и показывает успешность завершения или причину неудачи (не смогла открыть файл, прочитать элемент и т.д.).
2. Написать функцию, получающую в качестве аргументов имя `a` текстового файла неизвестной длины, имя файла `b` для вывода информации, символьные строки `s`, `t`, указатель на целое число `r`, и выводящую в файл `b` те строки файла `a`, которые состоят только из слов строки `s`. Словом называется последовательность символов, не содержащая пробельных символов, пробельным называется символ, содержащийся в строке `t`. Для быстрого поиска все слова в строке `s` размещаются в красно-черном дереве строк до начала работы с текстовыми файлами. Сами строки размещаются как объекты `std::unique_ptr<char[]>`. Каждая прочитанная из файла строка разбирается на слова, которые затем ищутся в красно-черном дереве. Функция возвращает в случае успеха количество выведенных в файл `b` строк в переменной `r`. Возвращаемое значение функции имеет тип `io_status` и показывает успешность завершения или причину неудачи (не смогла открыть файл, прочитать элемент и т.д.).

3. Написать функцию, получающую в качестве аргументов имя *a* текстового файла неизвестной длины, имя файла *b* для вывода информации, символьные строки *s*, *t*, *x*, указатель на целое число *r*, и выводящую в файл *b* строки файла *a*, заменяя каждое вхождение слова из строки *s* на соответствующее слово строки *x*; соответствие осуществляется по номеру слова в строке, если слов в строке *s* больше, чем в *x*, то недостающие слова полагаются равными пустым строкам, если же слов в строке *s* меньше, чем в *x*, то лишние слова в *x* игнорируются. Словом называется последовательность символов, не содержащая пробельных символов, пробельным называется символ, содержащийся в строке *t*. Для быстрого поиска все слова в строке *s* размещаются в AVL дереве пар строк (слово из строки *s*, являющееся ключом поиска, и соответствующее ему слово из строки *x*) до начала работы с текстовыми файлами. Сами строки размещаются как объекты `std::unique_ptr<char[]>`. Каждая прочитанная из файла строка разбирается на слова, которые затем ищутся в AVL дереве. Функция возвращает в случае успеха количество выведенных в файл *b* строк в переменной *r*. Возвращаемое значение функции имеет тип `io_status` и показывает успешность завершения или причину неудачи (не смогла открыть файл, прочитать элемент и т.д.).
4. Написать функцию, получающую в качестве аргументов имя *a* текстового файла неизвестной длины, имя файла *b* для вывода информации, символьные строки *s*, *t*, *x*, указатель на целое число *r*, и выводящую в файл *b* те строки файла *a*, в которых есть слово, удовлетворяющее условию, задаваемую строкой *x*, по отношению к слову в строке *s*. Словом называется последовательность символов, не содержащая пробельных символов, пробельным называется символ, содержащийся в строке *t*. Строка *x* может принимать значения "<", ">", "<=", ">=" – как в C++, "=" – равно, "<>" – не равно. Количество условий в строке *x* и слов в строке *s* должно совпадать, и, если их более одного, то для каждого слова из строки файла *a* применяется условие "ИЛИ". Например, для *s*="ab cd", *x*=">" подходят все слова, которые либо совпадают с "ab", либо лексикографически больше "cd". Функция возвращает в случае успеха количество выведенных в файл *b* строк в переменной *r*. Возвращаемое значение функции имеет тип `io_status` и показывает успешность завершения или причину неудачи (не смогла открыть файл, прочитать элемент и т.д.).
5. Написать функцию, получающую в качестве аргументов имя *a* текстового файла неизвестной длины, имя файла *b* для вывода информации, символьные строки *s*, *t*, *x*, указатель на целое число *r*, и выводящую в файл *b* те строки файла *a*, в которых все слова удовлетворяют условию, задаваемую строкой *x*, по отношению к слову в строке *s*. Словом называется последовательность символов, не содержащая пробельных символов, пробельным называется символ, содержащийся в строке *t*. Строка *x* может принимать значения "<", ">", "<=", ">=" – как в C++, "=" – равно, "<>" – не равно. Количество условий в строке *x* и слов в строке *s* должно совпадать, и, если их более одного, то для каждого слова из строки файла *a* применяется условие "И". Например, для *s*="ab cd", *x*="> <" подходят все слова, которые лексикографически больше "ab" и меньше "cd". Функция возвращает в случае успеха количество выведенных в файл *b* строк в переменной *r*. Возвращаемое значение функции имеет тип `io_status` и показывает успешность завершения или причину неудачи (не смогла открыть файл, прочитать элемент и т.д.).
6. Написать функцию, получающую в качестве аргументов имя *a* текстового файла неизвестной длины, имя файла *b* для вывода информации, символьные строки *s*, *t*, указатель на целое число *r*, и выводящую в файл *b* те строки файла *a*, в которых есть слово, совпадающее со словом в строке *s*. При этом в слове *s* символ "\_" соответствует одному любому

символу, а символы `"\_"` и `"\\"` соответствуют литеральным символам `"_"` и `"\"`. Комбинация `'\символ'` в строке  $s$ , где `'символ'` не равен `'_'` или `'\"`, не имеет специального значения и эквивалентна просто `'символ'`. Одиночный символ `'\"` в последней позиции строки  $s$  является ошибкой ввода. Словом называется последовательность символов, не содержащая пробельных символов, пробельным называется символ, содержащийся в строке  $t$ . Функция возвращает в случае успеха количество выведенных в файл  $b$  строк в переменной  $r$ . Возвращаемое значение функции имеет тип `io_status` и показывает успешность завершения или причину неудачи (не смогла открыть файл, прочитать элемент и т.д.).

7. Написать функцию, получающую в качестве аргументов имя  $a$  текстового файла неизвестной длины, имя файла  $b$  для вывода информации, символьные строки  $s$ ,  $t$ , указатель на целое число  $r$ , и выводящую в файл  $b$  те строки файла  $a$ , в которых есть слово, совпадающее со словом в строке  $s$ . При этом в слове  $s$  символ `"%"` соответствует 0 или более любым символам, а символы `"\%"` и `"\\"` соответствуют литеральным символам `"%"` и `"\"`. Комбинация `'\символ'` в строке  $s$ , где `'символ'` не равен `'%'` или `'\"`, не имеет специального значения и эквивалентна просто `'символ'`. Одиночный символ `'\"` в последней позиции строки  $s$  является ошибкой ввода. Достаточно рассмотреть следующие варианты в строке  $s$ :

- `"text%"` – соответствует всем словам, начинающимся с `"text"`,
- `"%text"` – соответствует всем словам, заканчивающимся на `"text"`,
- `"%text%"` – соответствует всем словам, включающим в себя `"text"`,
- `"text1%text2"` – соответствует всем словам, начинающимся с `"text1"` и заканчивающимся на `"text2"`.

Здесь `"text"`, `"text1"`, `"text2"` может содержать символы `'\%'` и `'\\'`, но не символ `'%'`. Словом называется последовательность символов, не содержащая пробельных символов, пробельным называется символ, содержащийся в строке  $t$ . Функция возвращает в случае успеха количество выведенных в файл  $b$  строк в переменной  $r$ . Возвращаемое значение функции имеет тип `io_status` и показывает успешность завершения или причину неудачи (не смогла открыть файл, прочитать элемент и т.д.).

8. Написать функцию, получающую в качестве аргументов имя  $a$  текстового файла неизвестной длины, имя файла  $b$  для вывода информации, символьные строки  $s$ ,  $t$ , указатель на целое число  $r$ , и выводящую в файл  $b$  те строки файла  $a$ , в которых есть слово, совпадающее со словом в строке  $s$ . При этом в слове  $s$  последовательность `'[n-m]'` ( $n$ ,  $m$  – символы) соответствует любому символу, имеющему код в диапазоне  $n \dots m$ , а символы `"\[", "\]"` и `"\"` соответствуют литеральным символам `"["`, `"]"` и `"\"`. Комбинация `'\символ'` в строке  $s$ , где `'символ'` не равен `'['`, `']'` или `'\"`, не имеет специального значения и эквивалентна просто `'символ'`. Одиночный символ `'\"` в последней позиции строки  $s$  является ошибкой ввода. Одиночный символ `'['`, `']'` без парного `']'`, `'['` является ошибкой ввода. Словом называется последовательность символов, не содержащая пробельных символов, пробельным называется символ, содержащийся в строке  $t$ . Функция возвращает количество таких строк или  $-1$ ,  $-2$  и т.д., если она не смогла открыть файл, прочитать элемент и т.д..
9. Написать функцию, получающую в качестве аргументов имя  $a$  текстового файла неизвестной длины, имя файла  $b$  для вывода информации, символьные строки  $s$ ,  $t$ , указатель на целое число  $r$ , и выводящую в файл  $b$  те строки файла  $a$ , в которых есть слово, совпадающее со словом в строке  $s$ . При этом в слове  $s$  последовательность `'[^n-m]'` ( $n$ ,  $m$  – символы) соответствует любому символу, имеющему код, не содержащийся в диапазоне  $n \dots m$ , а символы `"\[", "\]", "\^"` и `"\"` соответствуют литеральным символам `"["`, `"]"`, `"^"`

и "\". Комбинация '\символ' в строке  $s$ , где 'символ' не равен '[', ']', '^' или '\', не имеет специального значения и эквивалентна просто 'символ'. Одиночный символ '\' в последней позиции строки  $s$  является ошибкой ввода. Одиночный символ '[', ']' без парного ']', '[' является ошибкой ввода. Словом называется последовательность символов, не содержащая пробельных символов, пробельным называется символ, содержащийся в строке  $t$ . Функция возвращает в случае успеха количество выведенных в файл  $b$  строк в переменной  $r$ . Возвращаемое значение функции имеет тип `io_status` и показывает успешность завершения или причину неудачи (не смогла открыть файл, прочитать элемент и т.д.).