

## Требования к программам

1. Задачи оцениваются независимо в двух группах: задачи 1–5 и задачи 6–10.
2. Программа должна получать все параметры в качестве аргументов командной строки.
3. Аргументы командной строки для задач 1–2, 6–10:

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

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"`.

4. 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  $t$ .

Например, запуск

```
./a05.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. Результатом работы каждой функции является измененный файл (не выводится в `main`) и возвращаемое значение (выводится в `main`).
6. 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 (1–10),
- `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.

7. Возвращаемым значением всех функций является тип данных, объявленный, например, в файле `io_status.h`

```
typedef enum io_status_ {
    SUCCESS,          /* все хорошо */
    ERROR_OPEN,       /* ошибка открытия файла */
    ERROR_READ,       /* ошибка чтения файла */
    ERROR_PATTERN,    /* неверная строка поиска */

} io_status;
```

8. Пример функции `main` для задачи 1:

```
#include <stdio.h>
#include <time.h>
#include "io_status.h"

io_status task1 (const char *a, const char *b, const char *s, int *r);

int main (int argc, char *argv[])
{
    int task = 1;
    io_status ret;
    const char * f_in = 0;
    const char * f_out = 0;
    const char * s = 0;
    int res = 0;
    double t;

    if (argc != 4)
    {
        printf("Usage: %s <f_in> <f_out> <s>\n", argv[0]);
        return 1;
    }

    f_in = argv[1];
    f_out = argv[2];
    s = argv[3];
    t = clock();
    ret = task1 (f_in, f_out, s, &res);
    t = (clock() - t) / CLOCKS_PER_SEC;

    switch (ret)
    {
        case SUCCESS:
            printf ("%s : Task = %d Result = %d Elapsed = %.2f\n",
                    argv[0], task, res, t);
            break;
        case ERROR_OPEN:
            printf ("Can not open %s\n", f_out);
            return 1;
        case ERROR_READ:
            printf ("Can not read %s\n", f_out);
            return 2;
```

```

    case ERROR_PATTERN:
        printf ("Error in pattern %s\n", s);
        return 3;
    }

    return 0;
}

```

## Задачи

1. Написать функцию, получающую в качестве аргументов имя  $a$  текстового файла неизвестной длины, имя файла  $b$  для вывода информации, символьную строку  $s$ , указатель на целое число  $r$ , и выводящую в файл  $b$  те строки файла  $a$ , в которые входит строка  $s$ ; при этом символ '^' в начале строки  $s$  соответствует началу строки из файла  $a$ , символам '^' и '\', соответствуют последовательности символов '\^' и '\\', в строке  $s$ . Символ '^' не в первой позиции строки  $s$  не имеет специального значения и эквивалентен '\^'. 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. 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, символьную строку  $s$ , указатель на целое число  $r$ , и выводящую в файл  $b$  те строки файла  $a$ , в которые входит строка  $s$ ; при этом символ '\$' в конце строки  $s$  соответствует концу строки из файла  $a$ , символам '\$' и '\' соответствуют последовательности символов '\\$' и '\\', в строке  $s$ . Символ '\$' не в последней позиции строки  $s$  не имеет специального значения и эквивалентен '\\$'. 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. 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  $s$  и  $t$ , указатель на целое число  $r$ , и выводящую в файл  $b$  те строки файла  $a$ , в которые входит строка  $s$ ; при этом последовательность символов '\', '<' в строке  $s$  соответствует началу слова в строке из файла  $a$ , (словом называется последовательность символов, не содержащая пробельных символов, пробельным называется символ, содержащийся в строке  $t$ ). Строка  $s$  не содержит пробельных символов (т.е. является одним словом), а последовательность '\', '<' может быть только в начале строки  $s$ . Комбинация '\символ' в строке  $s$ , где 'символ' не равен '<', не имеет специального значения и эквивалентна просто 'символ'. Одиночный символ '\' в последней позиции строки  $s$  является ошибкой ввода. 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  $s$  и  $t$ , указатель на целое число  $r$ , и выводящую в файл  $b$  те строки файла  $a$ , в которые входит строка  $s$ ; при этом последовательность символов `'\','>'` в строке  $s$  соответствует концу слова в строке из файла  $a$ , (словом называется последовательность символов, не содержащая пробельных символов, пробельным называется символ, содержащийся в строке  $t$ ). Строка  $s$  не содержит пробельных символов (т.е. является одним словом), а последовательность `'\','>'` может быть только в конце строки  $s$ . Комбинация `'\символ'` в строке  $s$ , где `'символ'` не равен `'>'`, не имеет специального значения и эквивалентна просто `'символ'`. Одиночный символ `'\'` в последней позиции строки  $s$  является ошибкой ввода. 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  $s$  и  $t$ , 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  $t$ ). 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, символьную строку  $s$ , указатель на целое число  $r$ , и выводящую в файл  $b$  те строки файла  $a$ , которые совпадают со строкой  $s$ ; при этом символ `'.'` в строке  $s$  соответствует любому символу строки из файла  $a$ , символам `'.'` и `'\'` соответствуют последовательности символов `'\.'` и `'\\'` в строке  $s$ . 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. 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, символьную строку  $s$ , указатель на целое число  $r$ , и выводящую в файл  $b$  те строки файла  $a$ , которые совпадают со строкой  $s$ ; при этом символ `'?'` в строке  $s$  означает, что предыдущий символ строки  $s$  может учитываться 0 или 1 раз, символам `'?'` и `'\'` соответствуют последовательности символов `'\?'` и `'\\'` в строке  $s$ . Символ `'?'` в первой позиции строки  $s$  не имеет специального значения и эквивалентен `'\?'`. Несколько подряд идущих специальных символов `'?'` эквивалентны одиночному специальному символу `'?'`. 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. 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, символьную строку  $s$ , указатель на целое число  $r$ , и выводящую в файл  $b$  те строки файла  $a$ , которые совпадают со строкой  $s$ ; при этом символ `'+'` в строке  $s$  означает, что предыдущий символ строки  $s$  может учитываться

1 или более раз, символам '+' и '\ ' соответствуют последовательности символов '\+' и '\\ ' в строке *s*. Символ '+' в первой позиции строки *s* не имеет специального значения и эквивалентен '\+'. Несколько подряд идущих специальных символов '+' эквивалентны одиночному специальному символу '+'. 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. 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.).

9. Написать функцию, получающую в качестве аргументов имя *a* текстового файла неизвестной длины, имя файла *b* для вывода информации, символьную строку *s*, указатель на целое число *r*, и выводящую в файл *b* те строки файла *a*, которые совпадают со строкой *s*; при этом символ '\*' в строке *s* означает, что предыдущий символ строки *s* может учитываться 0 или более раз, символам '\*' и '\ ' соответствуют последовательности символов '\\*' и '\\ ' в строке *s*. Символ '\*' в первой позиции строки *s* не имеет специального значения и эквивалентен '\\*'. Несколько подряд идущих специальных символов '\*' эквивалентны одиночному специальному символу '\*'. 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. 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.).
10. Написать функцию, получающую в качестве аргументов имя *a* текстового файла неизвестной длины, имя файла *b* для вывода информации, символьную строку *s*, указатель на целое число *r*, и выводящую в файл *b* те строки файла *a*, которые совпадают со строкой *s*; при этом последовательность '[*n*-*m*]' (*n*, *m* – символы) в строке *s* соответствует любому символу строки из файла *a*, имеющему код в диапазоне *n*...*m*, символам '[', ']' и '\ ' соответствуют последовательности символов '\[', '\]' и '\\ ' в строке *s*. 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. 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.).