

Требования к программам

1. Программа должна получать все параметры в качестве аргументов командной строки.
2. Аргументы командной строки:

- 1) n – длина массива строк,
- 2) p – максимальное количество выводимых строк,
- 3) f_{in} – имя входного файла со строками,
- 4) s – строка.

Например, запуск

```
./a01.out 10 5 a.txt "abcd"
```

означает, что требуется прочитать массив строк длиной 10 из файла `a.txt`, выводить не более 5 строк массива, параметр–строка s равен `"abcd"`.

3. Ввод массива строк должен быть оформлен в виде функции. Строки текстового файла (без символа конца строки) считываются как элементы массива строк. Функция должна возвращать признак ошибки, если указанный файл не может быть прочитан, содержит меньшее количество данных, данные неверного формата или недостаточно памяти для чтения данных.
4. Решение задачи должно быть оформлено в виде функции, получающей в качестве аргументов массив, его длину и дополнительный параметр – строку. Получать в этой функции дополнительную информацию извне запрещается. Также в этой функции запрещается что-либо выводить на экран.
5. Программа должна содержать подпрограмму `print_array` вывода на экран массива длины не более p . Эта подпрограмма используется для **вывода исходного массива** после его инициализации, а также для вывода на экран результирующего массива. Подпрограмма получает массив, его длину и выводит на экран не более, чем p элементов массива, где p – параметр этой подпрограммы (аргумент командной строки). Каждый элемент массива должен **печататься на новой строке**.
6. Программа должна содержать подпрограмму `free_array`, получающую массив, его длину, и удаляющую всю выделенную под массив строк память.
7. Результатом работы каждой функции является измененный массив (выводится в `main`) и возвращаемое значение – его новая длина (выводится в `main`). Никакие функции не должны обращаться к элементам за новым концом массива.
8. Вывод результата работы функции в функции `main` должен производиться по формату:

```
printf ("New array:\n");  
print_array (a, res, p); /* вывод массива a длины res */  
printf ("%s : Task = %d Result = %d Elapsed = %.2f\n",  
        argv[0], task, res, t);  
free_array (a, res); /*удаление массива a длины res */
```

где

- `argv[0]` – первый аргумент командной строки (имя образа программы),
- `task` – номер задачи (1–8),
- `res` – возвращаемое значение функции, реализующей решение этой задачи,
- `t` – время работы функции, реализующей решение этой задачи.

Задачи

1. Написать функцию, получающую в качестве аргументов массив a строк, целое число n , являющееся длиной этого массива, и строку s , и выбрасывающую из массива все элементы, лексикографически меньшие строки s ; функция возвращает длину получившегося в результате массива (число пересылок элементов массива не должно превосходить $n + O(1)$).
2. Написать функцию, получающую в качестве аргументов массив a строк, целое число n , являющееся длиной этого массива, и строку s , и выбрасывающую из массива все элементы, которые входят в строку s ; функция возвращает длину получившегося в результате массива (число пересылок элементов массива не должно превосходить $n + O(1)$).
3. Написать функцию, получающую в качестве аргументов массив a строк, целое число n , являющееся длиной этого массива, и строку s , и выбрасывающую из массива все элементы, которые не имеют общих символов со строкой s ; функция возвращает длину получившегося в результате массива (число пересылок элементов массива не должно превосходить $n + O(1)$).
4. Написать функцию, получающую в качестве аргументов массив a строк, целое число n , являющееся длиной этого массива, и строку s , и выбрасывающую из массива все подряд идущие элементы, имеющие длину меньше, чем строка s , оставляя только первый элемент в таком участке; функция возвращает длину получившегося в результате массива (число пересылок элементов массива не должно превосходить $n + O(1)$).
5. Написать функцию, получающую в качестве аргументов массив a строк, целое число n , являющееся длиной этого массива, и строку s , и выбрасывающую из массива все подряд идущие элементы, у которых конец строки совпадает со строкой s , оставляя только первый элемент в таком участке; функция возвращает длину получившегося в результате массива (число пересылок элементов массива не должно превосходить $n + O(1)$).
6. Написать функцию, получающую в качестве аргументов массив a строк, целое число n , являющееся длиной этого массива, и строку s , и выбрасывающую из массива все подряд идущие элементы, которые не имеют общих символов со строкой s , оставляя только первый и последний элементы в таких участках; функция возвращает длину получившегося в результате массива (число пересылок элементов массива не должно превосходить $n + O(1)$).
7. Написать функцию, получающую в качестве аргументов массив a строк, целое число n , являющееся длиной этого массива, и строку s , и выбрасывающую из массива все элементы, имеющие соседа (слева или справа), лексикографически меньшего строки s ; функция возвращает длину получившегося в результате массива (число пересылок элементов массива не должно превосходить $n + O(1)$).
8. Написать функцию, получающую в качестве аргументов массив a строк, целое число n , являющееся длиной этого массива, и строку s , и выбрасывающую из массива все элементы, имеющие соседа (слева или справа), которые входят в строку s , причем прописные и соответствующие строчные латинские буквы считаются совпадающими; функция возвращает длину получившегося в результате массива (число пересылок элементов массива не должно превосходить $n + O(1)$).