

Напишите две функции, выполняющие одинаковые действия (в соответствии с указанным вариантом):

- использующую только строки в стиле C
- использующую строки типа string

В функциях разрешается использовать библиотечные функции из `<cctype>` и методы класса string, но запрещается(!) пользоваться функциями из `<cstring>`.

При работе со строками в стиле C не следует передавать в функцию размер строки или вычислять его внутри функции в отдельном цикле. В этом нет необходимости, т.к. все строки должны оканчиваться символом `'\0'` и можно работать с ними, учитывая это правило.

Если функция формирует новую строку, то она должна иметь вид:

- `char * foo(char *destination, const char *source, ...)`  
Для функции, работающей со строками в стиле C. Где destination – строка, в которую помещается результат, source – исходная строка (строки), возвращаемое значение – указатель на строку назначения.  
Строка destination должна быть передана в функцию извне, чтобы избежать выделения динамической памяти внутри функции (по примеру стандартных функций из `<cstring>`).
- `string bar(const string &source, ...)`  
Для функции, работающей с объектами string.  
Объект string сам управляет содержащейся в нём динамической памятью и имеет конструктор копирования. Поэтому результат можно вернуть из функции по значению.  
Объект string имеет метод `size()`, возвращающий размер строки, так что работать с символом `'\0'` не нужно.

Напишите функцию main для проверки ваших функций:

- Строки в стиле C следует разместить в динамической памяти.
- Длину строки (или строк) и сами символы строки считывать из файла, как строку в стиле C. В файле может быть сколько угодно строк (или пар строк, в зависимости от задания).
- Каждая строка описывается в файле двумя строчками: первая – количество символов в строке (выделяя память не забывайте про завершающий символ `'\0'`), вторая – символы строки.
- Для считывания самой строки удобнее всего использовать метод `getline()` класса `istream`.
- Объект string для демонстрации работы второй функции можно получить из строки в стиле C.
- Результаты работы программы вывести в поток `cout`.

Варианты:

1. Сформировать новую строку, удалив из исходной повторяющиеся пробелы (идущие подряд), а также пробелы в начале и в конце строки.
2. Сформировать новую строку, удалив из исходной строки все десятичные цифры.
3. Сформировать новую строку, удалив из исходной строки все буквы латинского алфавита.
4. Сформировать новую строку из символов, общих для двух исходных строк. В итоговой строке символы должны встречаться по одному разу, порядок символов не имеет значения.
5. Сформировать новую строку из символов двух исходных строк, которые НЕ являются для них общими. В итоговой строке символы должны встречаться по одному разу, порядок символов не имеет значения.

6. Сформировать новую строку из символов, которые в исходной строке повторяются более одного раза. В итоговой строке символы должны встречаться по одному разу, порядок символов не имеет значения.
7. Определить, сколько различных букв латинского алфавита содержится в строке.
8. Определить есть ли повторяющиеся символы в заданной строке. Функция должна возвращать true, если в строке есть повторяющиеся символы, false – в противном случае.
9. Определить, есть ли в заданной строке подряд стоящие одинаковые символы. Функция должна возвращать true, если такие символы в строке есть, false – в противном случае.
10. Сформировать новую строку, заменив в исходной строке все вхождения одного заданного символа на другой заданный символ.
11. Сформировать новую строку, заменив в исходной строке все прописные латинские буквы на строчные.
12. Сформировать новую строку, объединив две строки p и q по правилу: p[0], q[0], p[1], q[1], .... Исходные строки могут иметь разную длину, в этом случае «хвост» более длинной строки оставить без изменений.
13. Проверить, есть ли в двух заданных строках одинаковые символы. Функция должна возвращать true, если есть одинаковые символы, false – в противном случае
14. Сформировать новую строку, содержащую все латинские буквы, отсутствующие в исходной строке. Заглавные и строчные буквы не различаются. Буквы новой строки должны следовать в порядке возрастания их кодов ASCII.
15. Сформировать новую строку из двух исходных, исключив из первой все символы, встречающиеся во второй.
16. Сформировать новую строку из двух исходных, добавив в первую все символы, встречающиеся во второй и являющиеся десятичными цифрами.
17. Определить, есть ли повторяющиеся цифры в заданной строке. Функция должна возвращать true, если в строке есть повторяющиеся цифры, false – в противном случае
18. Сформировать новую строку, удалив из исходной строки все гласные буквы латинского алфавита.
19. Сформировать новую строку, состоящую из трёх чаще всего встречающихся символов исходной строки. Символы новой строки должны следовать в порядке возрастания их кодов ASCII.
20. Сформировать новую строку, содержащую все латинские буквы, присутствующие в двух заданных строках. Заглавные и строчные буквы не различаются. В итоговой строке буквы должны встречаться по одному разу и следовать в порядке возрастания их кодов ASCII.