

Основной преподаватель:
Смирнов Андрей Александрович

◀ Предыдущее занятие (../211099)

Следующее занятие ▶ (../211101)

Тема занятия: *Написание собственного класса "string"*
Набранные баллы: **275 / 1400**
Оценки: **5**

🟡 онлайн событие

дорешивание на 3 месяца

- 1. Вебинар
 - 1.1 Школа программистов.Онлайн



🏆 Результаты

🔴 Блиц-тест

окончен

- 1. Тестирование
 - 1.1 Блиц-тест

275 / 300

🏆 Результаты

🟢 Классная работа

дорешивание на 3 месяца

- 1. Задачи
 - 1.1 Класс String: введение
 - 1.2 A. Класс String: ввод и вывод
 - 1.3 B. Класс String: индексация и длина
 - 1.4 C. Класс String: конкатенация
 - 1.5 D. Класс String: умножение на число
 - 1.6 E. Класс String: сравнение
 - 1.7 F. Класс String: поиск подстроки
 - 1.8 G. Класс String: получение отдельных слов
 - 1.9 H. Структура set



- / 100

- / 100

- / 100

- / 100

- / 100

- / 100

- / 100

- / 100


2. Материалы

2.1 Материалы

2.2 Скриншоты с задачи "Ассоциативный массив"



Результаты

 (contests/252755/statements/)

Блиц-тест (дорешивание)

дорешивание на 3 месяца

1. Тестирование

1.1 Блиц-тест

?

Результаты

1. Задачи

Класс String: введение

0?

Создайте класс, который хранит внутри себя строку и её длину в отдельном целочисленном поле.

Добавьте следующие методы:

- конструктор по умолчанию, который создаёт пустую строку;
- конструктор, который создаёт объект из строки `char*`, копируя строку внутрь себя;
- деструктор;
- конструктор копирования;
- оператор присваивания.

Прототип итогового класса должен выглядеть примерно так:

```
class String {
    int len;
    char *str;

public:
    String();
    String(const String&);
    String(char *str);
    ~String();
    String& operator=(const String &);
};
```

При написании класса можно пользоваться функциями из библиотеки `cstring`.

A. Класс String: ввод и вывод

0?

Имя входного файла

стандартный ввод

Имя выходного файла

стандартный вывод

Ограничение по времени

2 секунды

Ограничение по памяти

64 мегабайта

Перегрузите в классе `String` методы ввода и вывода, которые позволят считать строку и вывести её на экран. Обратите внимание, что должна считываться вся строка до символа перевода строки, а не одно слово (токен).

В этой задаче введите строку с помощью класса `String` и выведите её обратно на экран.

Формат входных данных

С клавиатуры вводится одна строка длины не более 10^5 символов. Символы имеют ASCII-коды в диапазоне от 32 до 126.

Формат выходных данных

Требуется вывести на экран вводимую строку.

Примеры**входные данные**

```
We will rock you.
```

выходные данные

```
We will rock you.
```

Отправить на проверку

Выберите файл

C (GNU C 5.2) ▼

Отправить

Номер послылки ↕

Язык ↕

Время ↕

Результат ↕

Баллы

0 из 0

В. Класс String: индексация и длина**0?****Имя входного файла**

стандартный ввод

Имя выходного файла

стандартный вывод

Ограничение по времени

2 секунды

Ограничение по памяти

64 мегабайта

Добавьте в класс `String` метод `int String::length()` , который возвращает длину строки.
Перегрузите для класса операцию `[]` так, чтобы пользователь мог обращаться и менять элементы строки: `char& operator[] (int);`
С помощью указанных методов поменяйте местами первый и последний символы в строке.

Формат входных данных

С клавиатуры вводится одна строка длины не более 10⁵ символов. Символы имеют ASCII-коды в диапазоне от 32 до 126.

Формат выходных данных

Требуется вывести исходную строку, у которой поменяли местами первый и последний символ.

Примеры

входные данные	выходные данные
I'll be back	k'll be bacI

Отправить на проверку

Выберите файл

C (GNU C 5.2) ▾

Отправить

Номер отправки ↕	Язык ↕	Время ↕	Результат ↕	Баллы
0 из 0				

C. Класс String: конкатенация0?

Имя входного файла

стандартный ввод

Имя выходного файла

стандартный вывод

Ограничение по времени

2 секунды

Ограничение по памяти

64 мегабайта

Переопределите у класса операцию `+` так, что она будет объединять две строки в одну, склеивая их. Эта операция в программировании называется «Конкатенацией».

В этой задаче с клавиатуры вводятся пять строк. Объедините их в одну длинную строку, и замените каждый второй символ на символ нижнего подчёркивания «`_`», после чего выведите получившуюся строку на экран.

Формат входных данных

С клавиатуры вводятся 5 строк. Каждая из строк имеет длину не более 10000 символов, и состоит из символов с ASCII-кодами от 32 до 126.

Формат выходных данных

Требуется вывести объединенную строку, каждый второй символ которой заменен на «_».

Примеры

входные данные	выходные данные
This is a very secret message	T_i_i_a_v_r_s_c_e_m_s_a_e

Отправить на проверку

Выберите файл

C (GNU C 5.2) ▾

Отправить

Номер посылки↕	Язык↕	Время↕	Результат↕	Баллы
				0 из 0

D. Класс String: умножение на число

0?

Имя входного файла	стандартный ввод
Имя выходного файла	стандартный вывод
Ограничение по времени	2 секунды
Ограничение по памяти	64 мегабайта

Добавьте для класса `String` операцию умножения на беззнаковое целое число так, что при умножении строка дублируется количество раз, равное числу.

Например, `String("abc") * 5 → String("abcabcabcabcabc")` .

С помощью этой операции выведите на экран квадрат $N \times M$ символов «@», не используя циклов в главной функции.

Формат входных данных

С клавиатуры вводятся два натуральных числа N и M ($N \cdot M \leq 10^5$).

Формат выходных данных

Требуется вывести прямоугольник из символов «@» заданного размера.

Примеры

входные данные	выходные данные
5 2	

@@
@@
@@
@@
@@

Отправить на проверку

Выберите файл

C (GNU C 5.2)

Отправить

Номер послылки	Язык	Время	Результат	Баллы
----------------	------	-------	-----------	-------

0 из 0

Е. Класс String: сравнение

0?

Имя входного файла

стандартный ввод

Имя выходного файла

стандартный вывод

Ограничение по времени

2 секунды

Ограничение по памяти

64 мегабайта

Определите для класса `String` (строка) операции `==`, `!=`, `<`, `>`, `<=` и `>=`. Строки должны сравниваться в так называемом *лексикографическом порядке*. Это порядок, в котором строки расположены в словаре.

Формально, если сравниваются две строки a и b , то алгоритм сравнения выглядит так:

- если $a[0] < b[0]$, то $a < b$;
- если $a[0] > b[0]$, то $a > b$;
- если $a[0] == b[0]$, то алгоритм переходит к сравнению следующего символа.

Если все символы в строках совпали, то меньшей считается более короткая строка. Например, `"abc" < "abcd"`. Если в строках совпадают все пары символов и строки имеют одинаковую длину, то они считаются равными.

Вам с клавиатуры вводится две строки s_1 и s_2 и знак операции между ними. Проверьте истинность неравенства/равенства.

Формат входных данных

В первой и третьей строке вводятся строки s_1 и s_2 соответственно. Каждая из двух строк состоит и символов с ASCII-кодами от 32 до 126, а их длины не превышают 100 000 символов.

Во второй строке вводится одна из шести строк — `<`, `>`, `<=`, `>=`, `==`, `!=`.

Формат выходных данных

Требуется вывести «YES», если неравенство/равенство выполняется, или «NO», если нет.

Примеры

входные данные	выходные данные
abc < bc	YES
abcde == abcde	YES
abc > abc	NO

Отправить на проверку

Выберите файл

C (GNU C 5.2) ▼

Отправить

Номер послылки ↕	Язык ↕	Время ↕	Результат ↕	Баллы
------------------	--------	---------	-------------	-------

0 из 0

F. Класс String: поиск подстроки

0?

Имя входного файла

стандартный ввод

Имя выходного файла

стандартный вывод

Ограничение по времени

2 секунды

Ограничение по памяти

64 мегабайта

Даны две строки t и s . Проверьте, что t входит в s как подстрока и найдите позицию первого вхождения, если входит.

Решение этой задачи оформите в виде метода `int String::find(String t)`, которая вернёт позицию первого вхождения параметра t в строку, или вернет -1, если вхождений нет.

Формат входных данных

С клавиатуры вводятся две строки t и s . Длина каждой из строк не превышает 1000, строки состоят только из символов с ASCII-кодами от 32 до 126.

Формат выходных данных

Требуется вывести индекс первого вхождения строки t в s как подстроки, или -1, если его нет.

Примеры

входные данные	выходные данные
ab abc	0
axe taxes	1
a z	-1

Отправить на проверку

Выберите файл

С (GNU C 5.2) ▼

Отправить

Номер отправки ↕	Язык ↕	Время ↕	Результат ↕	Баллы
				0 из 0

G. Класс String: получение отдельных слов

0?

Имя входного файла

стандартный ввод

Имя выходного файла

стандартный вывод

Ограничение по времени

2 секунды

Ограничение по памяти

64 мегабайта

Перегрузите в классе `String` операцию `String String::operator() (int index)` так, что при её вызове из строки выделяется слово под номером `index`.

Например,

- `String("The quick brown fox jumps over the lazy dog")(0) → String("The");`
- `String("The quick brown fox jumps over the lazy dog")(2) → String("brown");`
- `String("The quick brown fox jumps over the lazy dog")(5) → String("over").`

В рамках данной задачи словом считается последовательность непробельных символов, которая с обеих сторон ограничена либо пробелами, либо концом/началом строки. При выборе слова с номером, которого не существует, операция должна возвращать пустую строку.

С помощью данной операции найдите лексикографически самое большое слово в строке.

Формат входных данных

С клавиатуры вводится строка, состоящая не более чем из 10^5 символов с ASCII-кодами от 32 до 126.

Формат выходных данных

Требуется вывести одно слово — самое лексикографически большое слово в исходной строке.

Примеры

входные данные	выходные данные
the quick brown fox jumps over the lazy dog	the

Отправить на проверку

Выберите файл

С (GNU C 5.2) ▼

Отправить

Номер отправки ↕	Язык ↕	Время ↕	Результат ↕	Баллы
------------------	--------	---------	-------------	-------

0 из 0

Н. Структура set

0?

Имя входного файла

стандартный ввод

Имя выходного файла

стандартный вывод

Ограничение по времени

2 секунды

Ограничение по памяти

64 мегабайта

Напишите программу, которая будет выполнять последовательность запросов вида ADD num, PRESENT num и COUNT (без параметра).

Выполнение каждого запроса вида ADD num должно добавлять элемент num во множество (если такой элемент уже есть, добавление ещё одной копии не изменяет множество), на экран при этом ничего не выводится.

При выполнении каждого запроса вида PRESENT num должно выдаваться сообщение «YES» или «NO» (большими буквами, в отдельной строке), соответственно тому, есть ли такой элемент во множестве; значение множества при этом не изменяется.

При выполнении каждого запроса вида COUNT должна выдаваться на экран в отдельной строке текущее количество различных элементов в множестве; значение множества при этом не изменяется.

Формат входных данных

В первой строке стандартного входного потока задано количество запросов N ($1 < N < 100000$), далее следуют N строк, каждая из которых содержит по одному запросу согласно описанного формата.

Значения чисел не превышают по модулю 10^8 .

Формат выходных данных

Выведите в отдельных строках результаты запросов PRESENT и COUNT; на запросы ADD ничего выводить не надо.

Примеры

входные данные	выходные данные
7	2
ADD 5	NO
ADD 7	YES
COUNT	3
PRESENT 3	
PRESENT 5	
ADD 3	
COUNT	

Отправить на проверку

Выберите файл

C (GNU C 5.2) ▼

Отправить

Номер посылки ↕	Язык ↕	Время ↕	Результат ↕	Баллы
				0 из 0