

Задача 1: Проверка палиндрома

Напишите программу, которая проверяет, является ли введённая строка палиндромом

Условие

Проверьте, читается ли строка одинаково слева направо и справа налево. Игнорировать пробелы и регистр. Работать только с символьным массивом.

Пример

Ввод: "A man a plan a canal Panama"

Вывод: Палиндром

Задача 2: Подсчет количества уникальных символов

Условие

Дана строка. Определите, сколько различных символов в ней встречается. Использовать только посимвольное сравнение.

Пример

Ввод: "success"

Вывод: 4 (символы: s, u, c, e)

Для каждого символа проверяйте, встречался ли он ранее, проходя все предыдущие позиции. Если не встречался — увеличивайте счётчик.

Задача 3: Удаление всех цифр из строки

Условие

Дана строка, содержащая буквы, пробелы и цифры. Удалите все цифры, оставив остальные символы.

Пример

Ввод: "C++17 is better than C89"

Вывод: "C++ is better than C"

С помощью двух индексов: один читает, другой записывает. Если символ не цифра — копируйте его в позицию записи.

Задача 4: Проверка, содержатся ли все символы одной строки в другой

Условие

Даны две строки A и B. Определите, содержатся ли **все символы** из B в A (порядок не важен).

Пример

```
Ввод:  
A = "information"  
B = "train"
```

Вывод: Да

Для каждого символа В ищите его в А с помощью посимвольного сравнения. Если хотя бы один не найден — ответ “Нет”.

Задача 5: Подсчет гласных и согласных

Условие

Считайте строку с клавиатуры и определите, сколько в ней гласных и согласных букв. Можно использовать английский алфавит. Все остальные символы (пробелы, знаки, цифры) не учитываются.

Пример

```
Ввод:  
Hello, world!
```

```
Вывод:  
Гласных: 3  
Согласных: 7
```

Считайте символы без учёта регистра (toupper() или tolower()), и проверяйте принадлежность к множеству гласных: А, Е, I, О, U, Y.

Задача 6: Шифр Цезаря

Условие

Реализуйте шифрование строки методом Цезаря. Каждая буква сдвигается в алфавите на заданное пользователем число позиций. Символы, не являющиеся буквами, не изменяются.

Сдвиг циклический: после Z идёт А.

Пример

```
Ввод:  
Текст: Hello, World!  
Сдвиг: 3
```

```
Вывод:  
Khoor, Zruog!
```

Используйте функции работы с символами:

- Если буква заглавная — диапазон 'A'..'Z';
- Если строчная — 'a'..'z';
- Сдвиг выполняйте по формуле:
- символ = (символ - base + shift) % 26 + base;

где `base` — 'A' или 'a' в зависимости от регистра, `shift` (сдвиг) — это целое число, показывающее, на сколько позиций по алфавиту нужно “сдвинуть” каждую букву.

Например

`shift = 3`

означает, что:

- $A \rightarrow D$
- $B \rightarrow E$
- $C \rightarrow F$
- $X \rightarrow A$
- $Y \rightarrow B$
- $Z \rightarrow C$

Задача 7: Сжатие строки

Условие

Реализуйте алгоритм RLE (Run-Length Encoding) — сжатие строк путем замены последовательностей одинаковых символов на символ и количество повторений.

Пример

Ввод: `aaabbbccccdd`

Вывод: `a3b2c1d5`

Задача 8: Автоматическая коррекция регистра в предложении

Условие

Напишите программу, которая корректирует регистр символов в тексте так, чтобы:

- Первая буква каждого предложения была заглавной,
- Остальные — строчные.

Пример

Ввод: `это пример. а вот второе предложение! третье?`

Вывод: `Это пример. А вот второе предложение! Третье?`

Определяйте границы предложений по символам `.`, `!`, `?`, а затем меняйте регистр первой буквы после них.

Задача 9: Поиск наибольшего общего подстрочного фрагмента

Условие

Даны две строки. Найдите самую длинную подстроку, которая встречается в обеих строках.

Пример

```
Ввод:  
str1 = "information"  
str2 = "formation"
```

```
Вывод: "formation"
```

Используйте двумерный динамический массив `dp[i][j]`, где `dp[i][j]` хранит длину совпадающей подстроки, заканчивающейся на `str1[i-1]` и `str2[j-1]`.

Задача 10: Инвертирование слов в предложении

Условие

Напишите программу, которая переворачивает каждое слово в строке, но сохраняет порядок слов.

Пример

```
Ввод: "C++ strings are powerful"  
Вывод: "++C sgnirts era lufrewop"
```

Разделите строку на слова, затем примените `std::reverse()` к каждому слову.

Задача 11: Нахождение всех подстрок длины k с уникальными символами

Условие

Дана строка и число `k`.

Необходимо вывести все подстроки длины `k`, в которых все символы различны.

Пример

```
Ввод:  
Строка: "abcabc"  
k = 3
```

```
Вывод:  
abc  
bca  
cab
```

Работайте с массивом символов (`char[]`). Для каждой подстроки длины `k` проверяйте, нет ли повторяющихся символов (например, двойным циклом по символам подстроки). Сдвигайте на один символ и повторяйте проверку до конца строки.