

Задание 3. ЯЗЫК АССЕМБЛЕРА. ОБРАБОТКА СИМВОЛЬНЫХ ДАННЫХ.

Требуется написать на ассемблере программу обработки текста.

Постановка задачи.

Дан текст – непустая последовательность не длиннее ста символов. Признаком конца ввода является точка, в сам текст точка не входит.

Проверить, удовлетворяет ли текст заданному условию. Если условие выполнено, преобразовать текст по одному правилу, в противном случае – по другому правилу. Преобразованный текст напечатать.

Проверяемое условие и правила обработки текста определяются конкретным вариантом задания.

Если введенная последовательность символов не является текстом, преобразовывать ее не нужно, а следует напечатать соответствующее сообщение.

Ввод текста, проверка условия, обработка текста и печать результата должны выполняться последовательно, отдельными частями программы.

Варианты.

Проверяемое условие.

- 1) Текст оканчивается прописной латинской буквой, которая больше в тексте не встречается. (с.к.)
- 2) Текст начинается цифрой и оканчивается цифрой, причем эти цифры различны.
- 3) Текст начинается и оканчивается латинской буквой.
- 4) Текст содержит не менее трех латинских букв.
- 5) Текст содержит равное количество прописных и строчных латинских букв.
- 6) Текст содержит равное количество прописных латинских и прописных русских букв.
- 7) Текст не содержит иных символов, кроме цифр и прописных русских букв.
- 8) В тексте больше латинских букв, чем цифр.
- 9) За каждой цифрой текста идет строчная латинская буква.
- 10) Количество строчных латинских букв в тексте равно сумме цифр в нем.
- 11) В тексте не встречаются одновременно цифра 0 и латинская буква O (в любом регистре).
- 12) Латинские буквы в тексте упорядочены по алфавиту.
- 13) Прописные русские буквы в тексте упорядочены по алфавиту.
- 14) В текст входят только строчные латинские буквы и цифры, причём буквы и цифры чередуются.

Первое правило преобразования.

- 1) Заменить каждую ненулевую цифру соответствующей ей строчной буквой латинского алфавита ($1 \rightarrow a$, $2 \rightarrow b$ и т.д.).
- 2) Заменить каждую строчную латинскую букву цифрой $N \bmod 10$, где N – порядковый номер буквы в алфавите.
- 3) Заменить каждую прописную русскую букву цифрой $N \bmod 10$, где N – порядковый номер буквы в алфавите.
- 4) Заменить каждую прописную латинскую букву следующей за ней по алфавиту, букву Z менять на A.
- 5) Заменить каждую прописную русскую букву следующей за ней по алфавиту, букву Я менять на A.
- 6) Заменить каждую строчную латинскую букву соответствующей прописной буквой, а прописную – строчной.
- 7) Заменить каждую прописную русскую букву соответствующей строчной буквой.

- 8) Заменить каждую латинскую букву симметричной ей в алфавите ($A \leftrightarrow Z, b \leftrightarrow y, \dots$).
- 9) Заменить каждую прописную русскую букву симметричной ей в алфавите ($A \leftrightarrow Я, Б \leftrightarrow Ю, \dots$).

Второе правило преобразования.

- 1) Перевернуть текст, не используя дополнительную память.
- 2) Удвоить каждую литеру текста. (Можно использовать дополнительную память.)
- 3) Удвоить каждую строчную латинскую букву текста. (Можно использовать дополнительную память.)
- 4) Циклически сдвинуть текст на три позиции влево. (с.к.)
- 5) Циклически сдвинуть текст на четыре позиции вправо. (с.к.)
- 6) Циклически сдвинуть текст на K (константа) позиций влево без использования дополнительной памяти, реализовав следующий алгоритм: перевернуть подмассив из первых K символов; перевернуть оставшийся подмассив; перевернуть весь текст.
- 7) Циклически сдвинуть текст на K (константа) позиций вправо без использования дополнительной памяти, реализовав следующий алгоритм: перевернуть подмассив из последних K символов; перевернуть оставшийся подмассив; перевернуть весь текст.
- 8) Поменять местами каждый символ на нечетной позиции и следующий за ним символ текста, если таковой имеется.
- 9) В каждой группе следующих подряд одинаковых литер оставить только одну из них. (с.к.)
- 10) Удалить символы в каждой второй позиции.
- 11) Удалить из текста все повторные вхождения его первой литеры. (с.к.)
- 12) Перенести в начало текста все входящие в него цифры с сохранением порядка их следования. (с.к.)

Требования к программе.

Нельзя использовать дополнительную память, размер которой зависит от длины текста (в частности, стек) во всех вариантах, кроме тех, где явно не указано обратное.

При реализации вариантов, помеченных "с.к.", требуется использовать строковые команды с префиксами повторения.

Проверку условия и выполнение преобразований текста следует реализовать в виде процедур. Глобальные переменные не использовать.

Интерфейс программы должен быть достаточно удобным: надо печатать приглашение для ввода, пояснять, какое условие проверяется, истинно ли оно и как текст преобразуется.

Рекомендации.

В некоторых вариантах удобно реализовать в качестве процедур или макросов проверку, является ли символ буквой, цифрой и т.п.

В процедуре преобразования текста может быть полезным иметь два указателя на позиции в тексте: откуда происходит чтение и куда идёт запись.

Максимально возможную длину текста описать в виде константы.