Homework 5: C-String Problem

Please do this homework at least three problems. Do more as a bonus.

(Problem 1 and problem 5 must be included)

1. Please write a function to process the string s="0123456789" and print the following output: (suggested problem)
(You should use the functions of the C-string only)

0123456789
1234567890
2345678901
3456789012
4567890123
5678901234
6789012345
7890123456
8901234567
9012345678

2. Please design a function to check whether the string parameter is a palindrome or not?
A string is a palindrome if it reads forward and backward in the same way. For example, the words: "refer", "madam" are palindrome.

Moreover you can consider one sentence as the input example to check whether this sentence is palindrome or not.

Case I: Use "character" as one unit.
Example: Was it a cat I saw? (delete all the space character)

Case II: Use "word" as one unit.
Example: Fall leaves as soon as leaves fall.

Case III: 中文 sentence:
Example: 一二三來三二一

Case IV: Chinese and English mixed sentence 中英文夾雜的句子
Dog 是不是 god ?

3. Write a program that reads one sentence from the keyboard. The delimiters are comma and space.

   (a). Count and print the total number of words in this sentence. **(5%)**

   Example:
   Input: This is a test and that is also a test
   Output: the total number of words: 10

   (b). Truncate the leading blank and backing blank of this sentence and print the result. **(5%)**
   Example:
   Input:_____This is a test._____
   Output:This is a test.

4. Please design a function to calculate the binary numbers addition:
   Example:

```
    101101110
+     1011111
---------------
    111001101
```

5. Please design a function translate binary number into corresponding decimal number:

   Example: $(1110111101)_2 = (957)_{10}$

5. Adding Large Integers: (suggested problem)

   In C++, the largest int value is 2147483647. So an integer larger than this number cannot be stored and processed as an integer. Similarly, if the sum or product of two positive integers is greater than 2147483647, The result will be incorrect. One way to store and manipulate large integers is to store each individual digit of the number in an array.

   Write a program that reads two positive integers of at most 40 digits and

prints the sum of the numbers. If the sum of the numbers has more than 40 digits, print the sum with an appropriate message.

```
   439549052177061179031076082376040 2
+    2563741836888159968824564772301 88
-----------------------------------------------------
   465186470545942778719321730099059 0
```

Your program must, at least, contain a function to read and store a number into an array and another function to print the sum of the numbers.

(Hint: Read numbers as strings and store the digits of the number in the reverse order.)

6 Look and Say:

The look and say sequence is defined as follows:
Start with any string of digits as the first element in the sequence. Each subsequence element is defined from the previous one by "verbally" describing the previous element. For example, the string 122344111 can be described as "one 1, two's 2, on 3, two 4's, three 1's". Therefore, the element that comes after 122344111 in the sequence is 1122132431. Similarly, the string 101 comes after 1111111111.

Input

The input consists of a number of cases. The first line gives the number of cases to follow. Each case consists of a line of up to 1000 digits.

Output

For each test case, print the string that follows the given string

Sample Input

```
3
122344111
11111111111
12345
```

Sample Output

```
1122132431
```

111
1112131415

7. The data compression algorithm RLE (Run Length Encoding) is based on the fact that a symbol within the data stream may be repeated many times. This repetitive sequence can be replaced by
   a. An Integer that declares the number of the repetitions.
   b. The symbol itself.

   Write a program that reads a string up to 100 characters and uses the RLE algorithm to compress it. Don't compress digits and characters that appear once.
   For example, the string fffmmmm1234eeeeeeeeeex should be compressed to 3f4m123410ex.

8. A Universal Product Code (UPC) barcode consists of 12 digits. The last digit is a check digit used for error detection. To calculate its value, we use the first 11 digits, as follows:

   a. Add the digits in the odd positions and multiply the result by three.
   b. Add the digits in the even positions to the previous result.
   c. Divide the result with 10. Subtract the remainder from 10, and that is the check digit. If the Subtraction gives 10(meaning that the remainder is 0), use 0 as check digit.

   For example, the check digit for the barcode 12345678901 is calculated as follows:
   a. 1+3+5+7+9+1=26. Multiplied by 3 gives 26*3=78.
   b. 2+4+6+8+0=20. Added to the previous result gives 98
   c. Check digit = 10 – (98 % 10) = 10 – 8 = 2

   Write a program that reads a UPC and verifies if the check digit is correct. The program should force the user to enter a valid UPC, meaning that the length of the string should be 12 and it must contain digits only.