ЗАДАЧА D. ДЕШИФРИРАНЕ

Петър и неговите приятели успели да разгадаят схемата за шифриране на съобщения на противниковата групировка. Всяко съобщение е редица от шифрирани знаци. При шифрирането, всеки знак се заменя с двоичен код, като шифриращите кодове изпълняват следните изисквания:

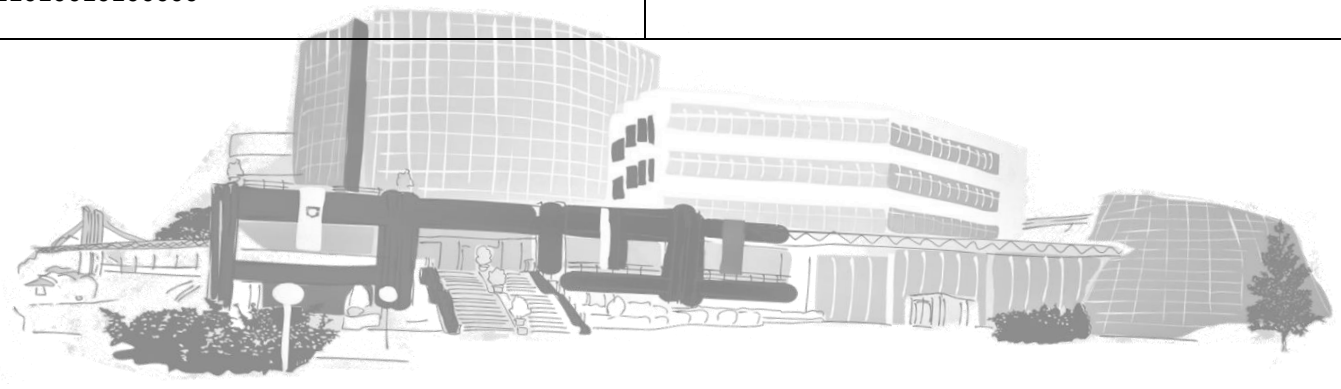
1. Кодовете на различните знаци са различни.
2. Кодовете може да са с различна дължина, но никой код не съдържа повече от 10 бита.
3. Кодът на нито един знак не е префикс (начало) на код на друг знак.

Шифриращите кодове са оформени в *кодова таблица*, която съдържа всички знаци, които могат да участват в съобщения и техните кодове. Например, с таблицата: **A:01; B:10; C:0010; D:0000**; може да се шифрират съобщения, съдържащи знаците {**A,B,C,D**}. Тази таблица отговаря на изискванията, а таблицата **A:01; B:10; C:010; D:0000**; не отговаря на изискванията, защото кодът на знака A е префикс на кода на знака C. Напишете програма, която по зададена кодова таблица и двоичен низ, който представлява коректно шифрирано съобщение, проверява дали кодовата таблица отговаря на изискванията и ако е така, декодира съобщението. Ако кодовата таблица не отговаря на изискванията, програмата трябва да изведе съобщение за грешка.

На **стандартния вход** ще бъдат зададени няколко тестови примера. Всеки тестов пример започва с ред, с броя N на елементите на кодовата таблица. Всеки един от следващите N реда е битов низ, който задава един елемент на кодовата таблица. Първите осем бита на низа са двоичното представяне на номера на знака в ASCII, а останалите K бита –кодът, с който се кодира този знак. На последния ред на примера е зададено коректно шифрирано съобщение – битов низ с дължина L. Тестовите примери завършват с ред, на който е зададена 0. В сила са следните ограничения $1 \le N \le 30$, $K \le 10$, $L \le 256$. В различните тестови примери се използват различни множества от знаци, а кодовете на един и същ знак в различните таблици не са непременно еднакви.

За всеки тестов случай програмата трябва да изведе по един ред на **стандартния изход** дешифрираното съобщение, ако кодовата таблица отговаря на изискванията, а в противен случай – съобщението **Wrong code table!**.

Примерен вход:	**Примерен изход:**
4 0100000101 0100001010 010000110010 010001000000 0110100100100000 4 0100000101 0100001010 01000011010 011010010000 011010010100000 0	ABBACD Wrong code table!



TASK D. DECRYPTING

Peter and his friends succeeded to unravel the encrypting scheme for messaging of the opposing group. Message is a sequence of encrypted characters. During the encrypting process each character is replaced by binary code. The encryption codes meet the following conditions:

1. Codes for different characters are different.
2. Codes could have different lengths, no grater then 10 bits.
3. No one code of a character is a prefix (beginning) of the code of another character.

Encrypting codes form coding table, which contains all characters that could be included in messages and their encryption codes. For example, with table: **A:01; B:10; C:0010; D:0000**; it is possible to encrypt messages that contain characters {**A,B,C,D**}. This table meet the conditions, but the table **A:01; B:10; C:010; D:0000**; does not meet them because the code of the character A is a prefix of the code of the character C. Write a program which for given coding table and binary string, which is correct encrypted message, check the correctness of the table and if the table is correct – decrypt the message. If the table is not correct the program has to print corresponding error message.

On the **standard input** few test cases will be given. Each test case starts with the number N of the elements of the coding table. Each of the next N lines is a binary string, which define one element of the coding table. First eight bits of this string is the binary presentation of the number of the character in ASCII table, and the remaining K bits – the encrypting code of the character. On the last line of the test case the encrypted message is given – binary string of length L. Test cases finish with line containing 0. The following limitations hold: $1 \le N \le 30$, $K \le 10$, $L \le 256$. In different test cases different sets of (printable) characters are used and codes of the same character in the different tables are not necessarily equal.

For each test case the program has to print on the separate line of the **standard output** the decrypted message if the coding table is meeting the conditions. Else – the program has to print **Wrong code table!.**

Example Input:	**Example Output:**
4 0100000101 0100001010 010000110010 010001000000 0110100100100000 4 0100000101 0100001010 01000011010 011010010000 011010010100000 0	ABBACD Wrong code table!