

题目描述：

TLV 编码是按[Tag Length Value]格式进行编码的，一段码流中的信元用 Tag 标识，Tag 在码流中唯一不重复，Length 表示信元 Value 的长度，Value 表示信元的值。

码流以某信元的 Tag 开头，Tag 固定占一个字节，Length 固定占两个字节，字节序为小端序。

现给定 TLV 格式编码的码流，以及需要解码的信元 Tag，请输出该信元的 Value。

输入码流的 16 进制字符串中，不包括小写字母，且要求输出的 16 进制字符串中也不要包含小写字母；码流字符串的最大长度不超过 50000 个字节。

输入描述：

输入的第一行为一个字符串，表示待解码信元的 Tag；

输入的第二行为一个字符串，表示待解码的 16 进制码流，字节之间用空格分隔。

输出描述：

输出一个字符串，表示待解码信元以 16 进制表示的 Value。

补充说明：

示例1

输入：31

32 01 00 AE 90 02 00 01 02 30 03 00 AB 32 31 31 02 00 32 33 33 01 00 CC

输出：32 33

说明：需要解析的信元的Tag是31，从码流的起始处开始匹配，Tag为32的信元长度为1（01 00，小端序表示为1）；第二个信元的Tag是90，其长度为2；第三个信元的Tag是30，其长度为3；第四个信元的Tag是31，其长度为2（02 00），所以返回长度后面的两个字节即可，即32 33。

```

1  import java.util.*;
2
3  public class Main {
4      public static void main(String[] args) {
5          Scanner cin = new Scanner(System.in);
6          String tag = cin.nextLine();
7          String[] g = cin.nextLine().split(" ");
8          int idx = 0;
9          boolean flag = false;
10         int n = g.length;
11         while (idx < n) {
12             ArrayList<String> list = new ArrayList<>();
13             String cur = g[idx];
14             idx++;
15             String a = g[idx];
16             idx++;
17             String b = g[idx];
18             idx++;
19             int m = Integer.parseInt(b + a, 16);
20             for (int i = 0; i < m; i++) {
21                 list.add(g[idx++]);
22             }
23             if (cur.equals(tag)) {
24                 StringJoiner res = new StringJoiner(" ");
25                 for (String s : list) {
26                     res.add(s);
27                 }
28                 flag = true;
29                 System.out.println(res.toString());
30                 break;
31             }
32         }
33         if (!flag) {
34             System.out.println("");
35         }
36     }
37 }
38

```