

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

展开

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。

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
const int maxn = 5e5 + 5;
```

```
int a[maxn];
```

```
int n, m;
```

```
int f(string s) {
```

```
    int ans = 0;
```

```
    if (s[1] >= '0' && s[1] <= '9') ans += s[1] - '0';
```

```
    else ans += s[1] - 'A' + 10;
```

```
    if (s[0] >= '0' && s[0] <= '9') ans += (s[0] - '0') * 16;
```

```
    else ans += (s[0] - 'A' + 10) * 16;
```

```
    return ans;
```

```
}
```

```
int main() {
```

```
    string s;
```

```
    cin >> s;
```

```
    n = f(s);
```

```
    while (cin >> s) {
```

```
        a[++a[0]] = f(s);
```

```
    }
```

```
    // for (int i = 1; i <= a[0]; ++i) cout<<a[i]<<' ';cout<<endl;
```

```
    for (int i = 1; i <= a[0]; i += a[i + 2] * 16 * 16 + a[i + 1] + 3) {
```

```
        if (a[i] == n) {
```

```
            for (int j = 1; j <= a[i + 2] * 16 * 16 + a[i + 1]; ++j) {
```

```
                // cout<<g(a[i + j + 2])<<endl;
```

```
                if (a[i + j + 2] >= 16) printf("%0X ", a[i + j + 2]);
```

```
                else printf("0%0X ", a[i + j + 2]);
```

```
        }
        break;
    }
}
return 0;
}
/*
31
32 01 00 AE 90 02 00 01 02 30 03 00 AB 32 31 31 02 00 F2 0A 33 01 00 CC
*/
```