L-IVE: Length Archiver

MikrosSoft (not to be confused with Microsoft) is an IT solution company which headquarter is in Medan. The company has a plan to develop a compression tool called L-IVE, short for Length ArchIVEr. L-IVE can be used to compress any type of data (text, music, video, image, etc.). It processes each **consecutive bytes of the same value** and present them with a 2 bytes data: byte count and byte value. Byte count represents the count, while byte value represents the value. Consider the following example.

```
  Original Data       Compressed Data
---------------       ---------------
      5 5 5 5  <->  4 5
            3  <->  1 3
  100 100 8 8 8  <->  2 100 3 8
5 5 4 4 4 5 5 5  <->  2 5 3 4 3 5
```

In the first example, there are 4 consecutive 5s, thus it can be represented as 4 5. In the last example, there are 2 consecutive 5s, 3 consecutive 4s, and followed by 3 consecutive 5s, thus the compressed data is 2 5 3 4 3 5. As you might have suspected, this compression might produce a longer data than the original, e.g., the second example.

Since a byte can only hold up to 256 different values (0..255). This compression technique can only compress up to 255 consecutive byte of the same value at a time. If it has more than 255 bytes, then it treats the remaining as a separate consecutive bytes. For example, a data containing 1000 bytes of consecutive 7s will be represented as 255 7 255 7 255 7 235 7. Note that 255 + 255 + 255 + 235 = 1000.

Your task in this problem is, given the original data, output the compressed data.

Input

Input contains a line with at least 1 and no more than 100,000 integers, each is separated by a single space. Each integer is between 0 and 255 (a byte), inclusive.

Output

Output in two lines. The first line contains an integer M denoting the number of bytes in the compressed data. The second line contains M integers representing the compressed data; each integer is separated by a single space.

Examples

input	Example #1

```
5 5 5 5
```

output

```
2
4 5
```

input   **Example #2**

3

output

2
1 3

---

input   **Example #3**

100 100 8 8 8

output

4
2 100 3 8

---

input   **Example #4**

5 5 4 4 4 5 5 5

output

6
2 5 3 4 3 5

End of Problem