

3383 Constant Irritation

Even for a single architecture — or even a single processor — there may be multiple assembler (or assembly) languages with different syntax rules, different directives (or “pseudo-ops”), and different mnemonics for the operation codes. Even the definition of integer constants can differ!

For example, one assembler for the Intel IA-32 architecture (that is, the x86, Pentium, and Xeon), uses a letter following a number to indicate the base, while another uses C/C++/Java style constants:

```
1f34h =    0x1f34    (a hex constant; C/C++/Java uses a leading 0x)
11001100b = 0xc0     (binary, but no direct C equivalent, so use hex)
377o =     0377     (octal; C/C++/Java uses a leading zero)
925 =      925     (decimal; C/C++/Java decimals must NOT begin with 0)
925d =     925     (also decimal)
```

And if that wasn't sufficient, the directives used to allocate and initialize storage for constants also differ:

```
db =      .byte      (one byte)
dw =      .word      (two bytes)
dd =      .long      (four bytes)
dq =      .quad      (eight bytes)
```

So using the assembler "A" (Intel IA-32 Architecture) we might write

```
db 1fh, 377o, 99d
```

but for assembler "B" (C/C++/Java style) we'd have to write

```
.byte 0x1f, 0377, 99
```

Each line of input for this problem will contain a directive that allocates and initializes storage, written in the style of assembler "A". Translate each such to the style required by assembler "B" and display it.

Input

Each input line contains, in order, the following items:

- zero or more whitespace characters (blanks and/or tabs)
- one of the directives 'db', 'dw', 'dd', or 'dq', in lowercase
- one or more whitespace characters
- one or more constants, separated from each other by a comma and zero or more whitespace characters (in any order)
- zero or more trailing whitespace characters
- the end of line character

There will be no more than 100 characters in any input line. The last input line (line with data) is followed by a line that contains only whitespace followed by an end of line character.

Output

For each input line you are to display one output line. The line is to contain the equivalent assembler “B” line for the input assembler “A” line. Specifically, it will contain, in order, the following items:

- exactly one tab character
- one of the directives ‘.byte’, ‘.word’, ‘.long’ or ‘.quad’, in lowercase letters, corresponding to the directive in the input line
- exactly one tab character
- one or more constants, separated from each other by a comma and then a blank. Each constant must be in the style supported by assembler “B”, and (if possible) use the same base as that given for the corresponding constant in the input line. For each binary constant in the assembler “A” form, use the hexadecimal form for assembler “B”.
- the end of line character

You are **not** to consider whether the given constant will fit in the storage allocated by the directive. Thus input like ‘db 99999 ’ is acceptable, even though that value will not fit in a byte.

Sample Input

```
dd    1,2 , 13h, 1010b , 12d, 377o
      dw  each , 0001b, 999999999999d
dq    1234h, 1200h
```

Sample Output

```
.long  1, 2, 0x13, 0xa, 12, 0377
.word  0xeac, 0x1, 999999999999
.quad  0x1234, 0x1200
```