

# Print Pretty

Your manager gave you a text file with many lines of numbers to format and print. For each row of **3** space-separated doubles, format and print the numbers using the specifications in the *Output Format* section below.

## Input Format

The first line contains an integer,  $T$ , the number of test cases.  
Each of the  $T$  subsequent lines describes a test case as **3** space-separated floating-point numbers:  $A$ ,  $B$ , and  $C$ , respectively.

## Constraints

- $1 \leq T \leq 1000$
- Each number will fit into a double.

## Output Format

For each test case, print **3** lines containing the formatted  $A$ ,  $B$ , and  $C$ , respectively. Each  $A$ ,  $B$ , and  $C$  must be formatted as follows:

1.  $A$ : Strip its decimal (i.e., truncate it) and print its hexadecimal representation (including the **0x** prefix) in lower case letters.
2.  $B$ : Print it to a scale of **2** decimal places, preceded by a  $+$  or  $-$  sign (indicating if it's positive or negative), right justified, and left-padded with underscores so that the printed result is exactly **15** characters wide.
3.  $C$ : Print it to a scale of exactly nine decimal places, expressed in scientific notation using upper case.

## Sample Input

```
1
100.345 2006.008 2331.41592653498
```

## Sample Output

```
0x64
      +2006.01
2.331415927E+03
```

## Explanation

For the first line of output,  $(100)_{10} \rightarrow (64)_{16}$  (in reverse,  $6 \times 16^1 + 4 \times 16^0 = (100)_{10}$  ).  
The second and third lines of output are formatted as described in the *Output Format* section.