

# Cube Numbers

For any positive integer  $n$ ,  $n$  can be represented as sum of other positive cube numbers ( $n=a_1^3+a_2^3+\dots+a_m^3$ ). Your task is to print the smallest  $m$ , where  $m$  is number of cube numbers used to form  $n$ , such that  $n=a_1^3+a_2^3+\dots+a_m^3$ . For example:

- $n=5, n=1^3+1^3+1^3+1^3+1^3$  ( $m=5$ )
- $n=8, n=2^3$  ( $m=1$ )
- $n=35, n=2^3+3^3$  ( $m=2$ )

**Note: My fastest time is 0.09s :).**

**Edit: My fastest time is 0.05s now lol**

**My Java solution is also accepted.**

## Input

Input consists of several test cases separated by new lines. Each test case consists of a positive integer, denoting the number of  $n$  ( $1 \leq n \leq 10^5$ ). Input is terminated by end of file (EOF).

It is guaranteed that total test case per input file is less than  $10^5$ .

**Note: For c++ users, you can use `while(scanf("%d",&n)!=EOF);` to read input until EOF.**

**Warning: large Input/Output data, be careful with certain languages!.**

## Output

For each case, print "Case #X: M", where  $X$  ( $1 \leq X \leq 10^5$ ) is the case number, and  $M$  is the minimum cube numbers used to form the integer  $n$ . There must be no trailing spaces at the end of printed lines, neither empty characters. Print a newline after each testcase.

## Example

**Input:**

1  
2  
5  
8  
35

**Output:**

Case #1: 1  
Case #2: 2  
Case #3: 5  
Case #4: 1  
Case #5: 2