Problem I
Itsy Bits

On most modern computers, unsigned numbers are stored in memory-aligned chunks of data, made up of *bits*. Depending on the size, one storage unit may be called a *word*, a *byte*, or even a *nybble*.

Each kind of storage takes exactly $b = 2^x$ bits of data, for some non-negative integer x. The smallest unsigned number that can be stored in a b-bit word is 0, and the largest is $2^b - 1$.

We are designing the storage for an embedded system where the maximum possible number in storage will be known upfront. Calculate the number of bits we should dedicate to it.

Input

- One line containing the largest number we need to store, n, ($1 \leq n \leq 10^{18}$).

Output

Output the number of bits we need to allocate to store n, which should be a power of two, followed by either "` bit`" or "` bits`" as appropriate.

Sample Input 1	Sample Output 1
`1`	`1 bit`

Sample Input 2	Sample Output 2
`37`	`8 bits`

Sample Input 3	Sample Output 3
`1100586419201`	`64 bits`

This page is intentionally left (almost) blank.