

# Problem 255: Mainframe Word Length

Difficulty: Easy

Author: Michael Warner, Denver, Colorado, United States

Originally Published: Code Quest 2025

## Problem Background

In computing, a "word" is the natural unit of data used in a processor design. It is a fixed number of bits (a "bit" is a single binary digit, 0 or 1). The bit length of registers, memory addresses, unsigned integers, instructions, and other quantities are typically the word length because it simplifies the computer design to keep these all consistent.

Modern computer architectures have settled on a *de facto* industry standard of 8-bit bytes, and a multiple of byte length for word length: usually 32-bit or 64-bit words. But earlier in computer history, the 8-bit byte hadn't been adopted, so there wasn't a "standard" word length. For the mainframes and minicomputers developed from the late 1940's to the mid-1970's, manufacturers chose word sizes that would look odd to us today.

Your team has been asked to help port an application from a modern 64-bit computer to run on some older architectures. You will need to see if the input values you're working with will fit within the target system's words.

## Problem Description

Your program will be provided a word length (in number of bits) followed by a set of unsigned integer values representing input data. An "unsigned integer" is an integer value that doesn't use one of its bits to indicate if it's positive or negative; as a result, they are never negative. Your program should print the maximum unsigned integer value that can be represented in the word length, followed by TRUE if every input data value will fit in the provided word length. Word lengths can be any value from 4 to 64, inclusive. The input data values will always fit within 64 bits.

You may wish to check your language's documentation to determine the appropriate data type for containing unsigned 64-bit integers.

## Sample Input

The first line of your program's input, received from the standard input channel, will contain a positive integer representing the number of test cases. Each test case will include two lines, as follows:

- A line containing a positive integer between 4 and 64 inclusive, representing the number of bits in a word in the legacy system architecture.

- A line containing between two and ten (inclusive) non-negative integers, separated by spaces and ranging from 0 to 18,446,744,073,709,551,615 inclusive, representing the input values to validate against the legacy architecture's word length.

```
2
12
1024 13 67 2999
36
3456 9999999 72165238905 5
```

## Sample Output

For each test case, your program must print a single line containing the following values, separated by a space:

- An integer representing the maximum unsigned integer value (in base-10) allowed by the legacy architecture's word length
- The word "TRUE" if every provided input value is less than or equal to the maximum unsigned integer value, or the word "FALSE" if at least one input value is greater than the maximum unsigned integer value, indicating it will not fit within the given word length.

```
4095 TRUE
68719476735 FALSE
```