

1. Implement the Miller-Rabin algorithm that checks whether a given number is prime in the range up to 2^{64}

https://en.wikipedia.org/wiki/Miller–Rabin_primality_test

<https://www.geeksforgeeks.org/primality-test-set-3-miller-rabin/>

https://www.tutorialspoint.com/c_standard_library/c_function_rand.htm

2. Write a program that sums Roman numerals. The input provides two Roman numbers separated by a space in single lines. The Roman numbers are in the range I ... M (in Arabic notation 1 ... 1000).

<https://www.baeldung.com/java-convert-roman-arabic>

The output on separate lines gives the result of adding the corresponding two numbers given in the input in Roman notation.

3. Most military codes are based on very large prime numbers. Your task is to break the code based on some prime number in the range (120,150). The entire string of characters is decrypted by one prime number. Each letter is encrypted with a three-digit number, the remainder of which when divided by the decryption number corresponds to the value of the character in the ASCII table (we only take into account capital letters A...Z). If the code cannot be read, it prints ILLEGIBLE

Input:

The first line gives the number of sequences $t < 25$, the next line gives the number of letters in the sequence $s < 25$, and the next line contains a set of three-digit numbers.

Output:

The output is a number that decrypts the string and the decoded word. If the code cannot be read, the program prints ILLEGIBLE

Input:

3

6

720 739 477 327 984 730

3

822 973 824

3

100 100 100

Output:

131 ATTACK

149 MOO

ILLEGIBLE